WHEN DOES COMPOSITIONAL STRUCTURE YIELD COMPOSITIONAL GENERALIZATION? A KERNEL THEORY.

Anonymous authors
Paper under double-blind review

ABSTRACT

Compositional generalization (the ability to respond correctly to novel combinations of familiar components) is thought to be a cornerstone of intelligent behavior. Compositionally structured (e.g. disentangled) representations are essential for this; however, the conditions under which they yield compositional generalization remain unclear. To address this gap, we present a general theory of compositional generalization in kernel models with fixed, compositionally structured representations, a tractable framework for characterizing the impact of dataset statistics on generalization. We find that these models are constrained to adding up values assigned to each combination of components seen during training ("conjunction-wise additivity"). This imposes fundamental restrictions on the set of tasks compositionally structured kernel models can learn, in particular preventing them from transitively generalizing equivalence relations. Even for compositional tasks that they can in principle learn, we identify novel failure modes in compositional generalization that arise from biases in the training data and affect important compositional building blocks such as symbolic addition and context dependence (memorization leak and shortcut bias). Finally, we empirically validate our theory, showing that it captures the behavior of deep neural networks (convolutional networks, residual networks, and Vision Transformers) trained on a set of compositional tasks with similarly structured data. Ultimately, this work provides a theoretical perspective on how statistical structure in the training data can affect compositional generalization, with implications for how to identify and remedy failure modes in deep learning models.

1 INTRODUCTION

Humans' understanding of the world is inherently compositional: once familiar with the concepts "pink" and "elephant," we can immediately imagine a pink elephant. Stitching together concepts in this way lets humans generalize far beyond our prior experience, allowing us to cope with unfamiliar situations and imagine things that do not yet exist (Lake et al., 2017; Frankland & Greene, 2020). Understanding the basis of compositional generalization in humans and animals, and building it into machine learning models, is a long-standing and historically vexing problem (Fodor & Pylyshyn, 1988; Battaglia et al., 2018; Lake & Baroni, 2018). While a wide range of studies have investigated the conditions under which compositionally structured (i.e. "disentangled") representations can be learned (Hinton et al., 2011; Higgins et al., 2017; Träuble et al., 2021; Whittington et al., 2023; Ren et al., 2023), it remains unclear when learning these representations is actually useful to a downstream neural network. Some work suggests that disentangled representations improve compositional generalization (Esmaeili et al., 2019; van Steenkiste et al., 2019). However, others challenge this view (Locatello et al., 2019; Schott et al., 2022). In general, it remains unclear why insights from some compositional tasks fail to generalize to others, and a systematic understanding of the relationships among compositional tasks remains elusive (Hupkes et al., 2020).

To make progress on this question, we focus on standard statistical learning, a fundamental basis for generalization that affects almost any machine learning model. Generalization in statistical learning depends on the similarities between different inputs. In compositionally structured representations, inputs that have components in common (say, a red circle and a blue circle) are more similar to each other than inputs that do not (say, a red circle and a blue square). As a result, the compositional structure of a task is reflected in its dataset statistics, influencing the model's generalization behavior.

To understand when and how statistical learning leads to successful compositional behavior, we developed a theory for the behavior of kernel models on compositional tasks. Kernel models are an important class of statistical models that are able to provide a simplified approximation to neural networks while maintaining analytical tractability (Jäkel et al., 2008; 2009). For example, kernel models accurately describe the behavior of learning and fine-tuning in neural networks under certain conditions (Jacot et al., 2018; Malladi et al., 2023). More broadly, they provide a tractable framework for characterizing the impact of dataset statistics on generalization (Canatar et al., 2021a;b).

Despite their broad relevance and relative simplicity, it has remained unclear how even kernel models generalize on compositional tasks (though see Abbe et al., 2023; Lippl et al., 2024, see Section 2). To address this gap, we present a theory of compositional generalization in kernel models. We define a general class of "compositionally structured representations" and a general family of compositional tasks with no constraints on the input-output mapping (ensuring broad applicability of our theory). We then theoretically characterize the compositional behavior of compositionally structured kernel models trained on such tasks, and go beyond kernel models to empirically validate our theory in several relevant deep neural network architectures. Our specific contributions are as follows:

- In Section 4, we show that kernel models with compositionally structured inputs are constrained to summing up values implicitly assigned to each component or combination of components seen during training. We call such computations "conjunction-wise additive."
- In Section 5, we then characterize how representational geometry determines whether kernel models will generalize successfully on conjunction-wise additive compositional tasks, highlighting two important failure modes (memorization leak and shortcut bias). This highlights that disentangled representations are not sufficient for downstream compositional generalization (even on conjunction-wise additive tasks) and explains why.
- Finally, in Section 6, we validate our theory in several deep neural network architectures, showing that it captures their behavior on conjunction-wise additive tasks.

Overall, we take a step towards a general theory of compositional generalization. Our theory systematically clarifies the compositional generalization behavior of deep networks and lays a foundation for the design of new learning mechanisms that overcome their limitations.

2 Related work

Compositional generalization. Compositionality is an important theme in both human and machine reasoning problems, including visual reasoning (Lake et al., 2015; Johnson et al., 2017; Schwartenbeck et al., 2023), language production (Hupkes et al., 2020), and rule learning (Ito et al., 2022; Abdool et al., 2023). Recent breakthroughs have led to massive improvements in models' compositional capacities, but in some cases, these models still fail spectacularly (Srivastava et al., 2023; Lewis et al., 2023; West, 2023). Attempts to improve compositional generalization often leverage meta-learning (Mitchell et al., 2021; Wu et al., 2023; Lake & Baroni, 2023) or modular architectures (Andreas et al., 2017), in the hopes that different modules will specialize for different components. Constraining a network's compositional function can guarantee modular specialization and compositional generalization (Lachapelle et al., 2023; Wiedemer et al., 2023a;b; Schug et al., 2024). However, these networks are extremely limited in the tasks they can learn (as we will show below) and end-to-end training of modular architectures without such constraints often does not result in the desired specialization (Bahdanau et al., 2019; Mittal et al., 2022; Jarvis et al., 2023).

Kernel regime. When using gradient descent, ridge regression, or similar learning algorithms to train the readout weights of a model $f_w(x) = w^T\phi(x)$, it can be shown that the resulting behavior of that model can be described in terms of the kernel $K(x, x') = \phi(x)^T\phi(x')$ induced by the representation $\phi(x)$ (Schölkopf, 2000). We will refer to such models as kernel models. Notably, when trained on a dataset $\{(x_i, y_i)\}_{i=1}^n$, a kernel model can be written in its "dual form" as

$$f_a(x) = \sum_{i=1}^n a_i K(x, x_i), \text{ for inferred dual coefficients } a \in \mathbb{R}^n. \quad (1)$$

Prior work has shown that neural networks with large initial weights or wide architectures are well approximated by gradient descent on a model with a fixed representation (a kernel model) (Jacot et al., 2018). This is called the "kernel regime," in contrast to the feature-learning regime, which is brought forth, for example, by small initial weights or small width (Chizat et al., 2019).

Norm minimization. Gradient descent, ridge regression, and neural networks in the kernel regime learn the readout weights that describe the training data with minimal ℓ_2-norm (Soudry et al., 2018; Gunasekar et al., 2018; Ji et al., 2020). Norm minimization is a standard theoretical framework for analyzing how representational geometry and dataset statistics influence generalization (Canatar et al., 2021b; 2023) and we here apply it to the compositional task setting. This is similar to Lippl et al. (2024) who characterize model behavior on a specific compositional task (transitive ordering) and Abbe et al. (2023) who characterize the inductive bias of norm minimization for inputs with binary components in the limit of infinite components. Compared to these prior works, we analyze a broader range of compositional tasks and derive exact constraints for finite numbers of components.

Memorization leak. Machine learning models sometimes memorize (parts of) their training data instead of learning a generalizable rule (Zhang et al., 2020; Dasgupta et al., 2022). This may improve generalization on long-tailed data (Feldman, 2020), but often impairs out-of-distribution performance (Elangovan et al., 2021). We find that models partially memorize their training data even when they extract the correct rule. Jarvis et al. (2023) analyze a related phenomenon in deep linear networks.

Shortcut learning. Shortcut learning refers to models exploiting spurious correlations between certain features of the input data and the target (Shah et al., 2020; Nagarajan et al., 2021). This substantially impacts their performance out of distribution (where those correlations may not hold) (Geirhos et al., 2020). We theoretically analyze how shortcut biases impact compositional generalization.

3 Model and Task Setup

3.1 Task Space

We consider an input x representing a set of underlying components $z = (z_c)_{c=1}^{C}$, where each component $z_c \in Z_c$ is drawn from a discrete, finite set of possible components that is fixed across all inputs. For example, x could be a simple concatenation of one-hot representations of all components (Fig. 1a). More generally, we consider a broad range of possible representations x, but make a specific assumption about how their trial-by-trial similarity relates to the underlying components z:

Definition 3.1. A representation x is "compositionally structured" iff its kernel $K(x, x') = x^T x'$ only depends on the number of components that are identical between z and z', where z and z' are the components represented by x and x' respectively.

In particular, the multi-hot representation described above is compositionally structured, but we will see below that this concept captures a much richer set of representations as well. The target, $y \in \mathbb{R}$, is given by an arbitrary function of z, ensuring that our framework is agnostic to the underlying compositional structure. After training models on certain combinations of components $Z^{\text{train}} \subset Z = \prod_{c=1}^{C} Z_c$, we assess generalization on all other combinations $Z^{\text{test}} := Z \setminus Z^{\text{train}}$.

Our theory characterizes constraints on model behavior for arbitrary tasks within this family, regardless of the ground-truth input-output mapping. Below we describe three example tasks that represent important building blocks for compositional reasoning in machine learning and cognitive science. We will focus on these tasks in the main text, to illustrate our general theory and analyze its consequences in more detail. In Appendix E, we describe additional tasks captured by our theory.

Symbolic addition. Many tasks involve inferring a magnitude associated with different underlying components (e.g. adding handwritten digits) (Lorenzi et al., 2021; Sheahan et al., 2021). We consider two components (z_1, z_2) with unobserved assigned values $v_1(z_1)$ and $v_2(z_2)$. The target is the sum of those values: $y = v_1(z_1) + v_2(z_2)$. After sufficient exposition to individual items, a model with an additive structure can generalize to novel combinations of items. In particular, we consider nine input elements $[-4], [-3], \ldots, [4]$ with associated values $-4, -3, \ldots, 4$ and training sets containing all pairs where at least one component is equal to a certain subset of values $\mathcal{W}$ (given by the rows and columns in the training set in Fig. 1b). We vary the size of this subset (i.e. the number of rows/columns) as well as whether the task requires only interpolation or also extrapolation.

Context dependence. The relevance of different stimuli often depends on our current context. Taking this into account is a crucial aspect of cognition in humans and animals (Bouton et al., 1999; Taylor & Ivry, 2013; Parker & Hollister, 2014; Ito et al., 2022). We therefore consider a task with three input components $(z_{\text{co}}, z_{f1}, z_{f2})$. The context (z_{co}) has two possible values specifying whether z_{f1} or z_{f2}

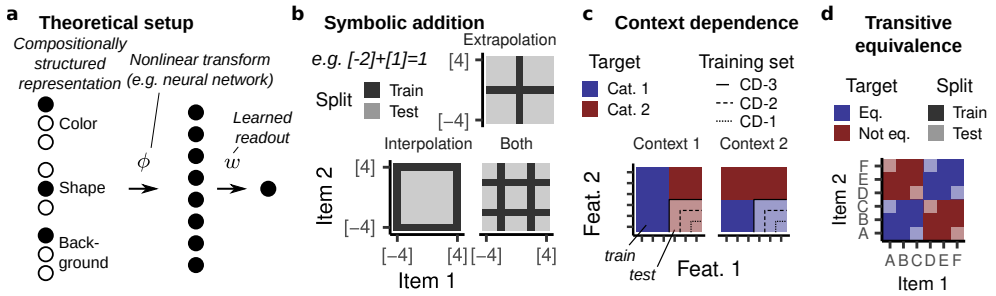

Figure 1: **a**, Theoretical setup: a disentangled representation of the input, followed by a nonlinear transform ϕ and a learned linear readout w. **b**, Example training sets for symbolic addition. The grid represents nine components with associated values $-4, -3, \ldots, 4$. The training set consists of certain rows and columns of this grid. **c**, Context dependence. In context 1, feat. 1 determines the category; in context 2, feat. 2 determines the category. The training sets leave out different subsets of the lower right orthant. **d**, Transitive equivalence: six items are split up into two arbitrary equivalence classes (e.g. A,B,C and D,E,F) and generalization requires transitive inference over equivalence classes.

determine the response. Both features have six possible values which are split up in two categories. If the model has learned this context dependence, it should be able to generalize to novel feature combinations. We evaluate on the subset of data for which z_{f1} indicates Cat. 2 and z_{f2} indicates Cat. 1. In the most extreme generalization test (*CD-3*), we leave out the entire subset; in easier versions, we leave out combinations of two or one of those features (*CD-2* and *CD-1*) (Fig. 1c).

Transitive equivalence. Relational reasoning is an important example of compositional generalization and often involves extending learned relations to new item combinations (Halford et al., 2010; Battaglia et al., 2018). Given an unobserved (and arbitrary) equivalence relation, the task is to determine whether two presented items (z_1, z_2) are equivalent. The model should generalize to novel item pairs using transitivity ($A = B$ and $B = C$ imply $A = C$) (Fig. 1d). This is an important instance of relational cognition (often studied as "associative inference" in cognitive science (Schlichting & Preston, 2015; Spalding et al., 2018)). Although prior work has found that kernel models often successfully generalize on a transitive *ordered* relation ($A > B$ and $B > C$ imply $A > C$, Lippl et al., 2024), the behavior of kernel models on equivalence relations has remained unclear.

3.2 Models

Our theory characterizes kernel models with a compositionally structured input $x \in \mathbb{R}^d$ that apply a transform $\phi(x) \in \mathbb{R}^h$ and learn a linear readout, $f_w(x) := w^T \phi(x)$, using gradient descent with initial weights $w_0 = 0$ (Fig. 1a). For ϕ, we consider a neural network with random weights (in the infinite-width limit; see Appendix A.3). This captures the random feature model studied by Abbe et al. (2023) and training via backpropagation in the kernel regime.

4 Kernel models are conjunction-wise additive

Our primary theoretical contribution is to characterize the full range of compositional computations that can be implemented by kernel models with compositionally structured inputs. We find that even though these models can learn arbitrary training sets, their test set behavior is restricted to adding up values for each conjunction (combination of components) seen during training. We call this motif "conjunction-wise additivity." Below, we formally state our finding and explain its implications.

4.1 Random networks yield compositionally structured representations

We first note that a linear readout of the multi-hot input constrains the model to adding up a value for each component ("component-wise additivity"): $f(x) = \sum_{c=1}^{C} f_c(z_c)$. Although models of this class perfectly generalize on component-wise additive tasks, such as symbolic addition, they are incapable of even learning the *training data* (much less generalizing) on tasks that are not component-wise

additive. In particular, these models cannot learn context-dependent computations or equivalence relations — both fundamental instances of compositional reasoning.

The nonlinear transform $\phi(x)$ can overcome this constraint; in particular, multi-layer nonlinear neural networks can learn arbitrary training data (in the infinite-width limit) (Hornik et al., 1989; Cybenko, 1989; Rigotti et al., 2013). In general, the wide range of possible network architectures makes it difficult to derive general statements about their representations, but in this case we can take advantage of the fact that $\phi(x)$ will also be compositionally structured:

Proposition 4.1. *For a random weights neural network ϕ with a compositionally structured input x in the infinite-width limit, $\phi(x)$ is also compositionally structured.*

Proposition 4.1 holds because in the infinite-width limit, the kernel of random weight neural networks depends only on the input kernel (Cho & Saul, 2009, see Appendix A.3). We now consider the constraints that compositional structure imposes on the models' generalization behavior.

4.2 Compositionally structured kernel models are conjunction-wise additive

We find that any kernel model with a compositionally structured representation is constrained to be conjunction-wise additive. To formally state this finding, we define, for each $z \in Z$, the set of conjunctions for which z overlaps with some element in the training set $z^{\text{tr}} \in Z^{\text{train}}$,

$$\text{Conj}(z|Z^{\text{train}}) := \left\{ J \subseteq \{1, \ldots, C\} | \exists z^{\text{tr}} \in Z^{\text{train}} : \forall c \in J : z_c = z_c^{\text{tr}} \right\}. \tag{2}$$

Theorem 4.2. *For any kernel model f with a compositionally structured representation, we can find conjunction-wise functions $f_J : \prod_{c \in J} Z_c \to \mathbb{R}$, where $J \subseteq \{1, \ldots, C\}$, such that for any input $x \in \mathbb{R}^d$ representing components $z \in Z$, the model response is given by*

$$f(x) = \sum_{J \in Conj(z|Z^{train})} f_J(z_J), \quad z_J := (z_c)_{c \in J}. \tag{3}$$

This means that for a given test input, these models add up a value for each partial conjunction (or single component) that was seen during training. We call this computation "conjunction-wise additive." In particular, for inputs with two components, the model's behavior on the training set can be expressed as $f(x) = f_1(z_1) + f_2(z_2) + f_{12}(z_1, z_2)$. Thus, $f_{12}(z_1, z_2)$ enables the model to learn arbitrary training data. However, if x represents $z \in Z^{\text{test}}$, the training set does not contain any input with the conjunction (z_1, z_2) and so the model is constrained to be component-wise additive: $f(x) = f_1(z_1) + f_2(z_2)$. For inputs with more than two components, a conjunction-wise additive computation additionally encodes partial conjunctions that it has seen before, e.g.: $f(x) = f_1(z_1) + f_2(z_2) + f_3(z_3) + f_{12}(z_1, z_2)$ if the training set contains the conjunction (z_1, z_2).

4.3 Conjunction-wise additivity constrains the tasks kernel models can solve

Theorem 4.2 implies that compositionally structured kernel models can only solve task that can be expressed in conjunction-wise additive terms. This highlights a fundamental computational restriction. Intriguingly, these restrictions are not caused by an architectural constraint, as the model can learn arbitrary training data (at least with a nonlinear transformation ϕ, see Section 5.1). Rather, we highlight restrictions on how models – without any architectural constraints – can *generalize* (Zhang et al., 2021). We now spell out the consequences of these restrictions.

First, for inputs with two components, we noted above that the model's behavior on the test set is component-wise additive. A striking consequence of this is that compositionally structured models can only generalize on component-wise additive tasks. In particular, this implies that these models cannot generalize on transitive equivalence — a fundamental instance of relational reasoning. Intriguingly, transitive ordering – a superficially similar task – can be solved by a kernel model. This highlights the importance of a formal perspective on compositional tasks.

More generally, to see whether a kernel model can, in principle, generalize on a task with more than two input components, we must 1) identify the conjunctions seen during training, $\text{Conj}(z|Z^{\text{train}})$, for each $z \in Z^{\text{test}}$, and 2) determine whether the target can be written as a conjunction-wise sum. For example, for context dependence, for all new test inputs, we have seen the context-feature

conjunctions 12 and 13, but not the feature-feature conjunction 23. As a result, model behavior on a test trial x representing the components $z_{\text{co}}, z_{f1}, z_{f2}$ can be written as $f(x) = f_1(z_{\text{co}}) + f_2(z_{f1}) + f_3(z_{f2}) + f_{12}(z_{\text{co}}, z_{f1}) + f_{13}(z_{\text{co}}, z_{f2})$. Importantly, this conjunction-wise additive function can encode context dependence: $f_{12}(z_{\text{co}}, z_{f1})$ can encode the target when the context z_{co} indicates the z_{f1} as relevant and is zero otherwise; $f_{13}(z_{\text{co}}, z_{f2})$ works in the opposite way. Thus, the model can, in principle, identify a set of weights that would allow it to generalize correctly.

This example illustrates that conjunction-wise additivity tells us both *whether* kernel models with compositionally structured representations can solve a certain task, and also *how* they solve it. As we noted, these insights immediately transfer to any model with compositionally structured representations that is trained or fine-tuned in the kernel regime. Because our theory determines the compositional computations emerging from dataset statistics, it is also likely to shed light on other learning models, including feature-learning deep neural networks (see Section 6).

Lastly, our theory tells us how to make a task non-additive: in Appendix E.2.3, we describe such a modification to context dependence that tests generalization to novel context-feature conjunctions. By helping us design hard benchmark tasks that are not solvable by kernel models, our framework grounds research into learning mechanisms that can implement other compositional computations.

In this section, we considered compositionally structured inputs. This is an idealization of practically relevant scenarios, where inputs will usually not be perfectly compositionally structured (even when these inputs are extracted from pretrained models). Our theory does not directly apply to these non-perfect cases. However, our results imply that even in the best-case scenario of perfectly compositionally structured inputs (e.g. a simple multi-hot input), compositional generalization in linear readout models faces fundamental restrictions. To further examine how these insights may translate to non-compositionally structured representations, we theoretically prove that the same limitation applies to randomly sampled representations that are only compositionally structured in expectation (Proposition A.2). We also empirically investigate compositional generalization in disentangled representation learning models trained on the DSprites dataset (Locatello et al., 2019), finding that, when averaged across randomly sampled task instances, these models are also limited to conjunction-wise additive computations (Appendix C). This indicates that the generalization class we have highlighted in this section may be relevant beyond our theoretical setting.

5 Representational geometry still impacts generalization on conjunction-wise additive tasks in kernel models

Conjunction-wise additivity only determines whether kernel models can, in principle, solve a certain task. However, model generalization also depends on whether the model identifies the correct conjunction-wise function. This depends on the model's representational geometry and the dataset statistics. As noted in Section 2, the behavior of kernel models depends on their representation $\phi(x)$ only through the induced kernel $K_\phi(x, x')$. In this section, we first investigate how different choices in network architecture influence its induced kernel (Section 5.1). Then, we characterize task performance on symbolic addition and context dependence across the full range of compositionally structured representations (Section 5.2).

5.1 Overlap salience characterizes compositional representational geometry

To characterize compositionally structured representations in this space, we introduce a new metric: representational salience. We formally define this metric in Appendix B.1. Intuitively, for $k = 1, \ldots, C$, the salience $S(k; C)$ measures the unique contribution of the subpopulation representing a conjunction of c components. Further, $S(k; C)$ is normalized so that all saliences sum up to one. For example, the multi-hot input only encodes single components and therefore $S(1; C) = 1/C$ and, for all $k > 1$, $S(k; C) = 0$. Note that the model can only use conjunctions whose salience is nonzero, e.g. the multi-hot input is constrained to a component-wise sum.

$S(c; C)$ is computed from the representational similarities $K(z, z')$. It is useful because unlike those similarities, it makes immediately apparent how strongly different conjunctions are encoded (Appendix B.2). Further, it reduces the set of $C + 1$ similarities characterizing the space of compositionally structured representations to $C - 1$ free parameters: It removes the overall magnitude of the representation (by normalizing) and the baseline activity (by excluding the similarity between inputs

with no overlap). This choice reflects the fact that both of these parameters often have negligible impact on model behavior (Appendix B.3).

We now analyze how $S(c; C)$ evolves over different layers of a random network, using an input with three components as an example (Fig. 2; see Appendix B.4 for other values of C). At the input stage, only individual components are represented and so $S(2; 3)$ and $S(3; 3)$ are zero. As the network gets deeper, these saliences increase, whereas $S(1; 3)$ decreases. In particular, because $S(3; 3) > 0$ for all of these networks, the resulting representations can learn arbitrary training data (using the full conjunction). Eventually, the salience of the intermediate conjunction $S(2; 3)$ decreases again, whereas $S(3; 3)$ continues to increase. Indeed, for ReLU networks, we prove the following statement:

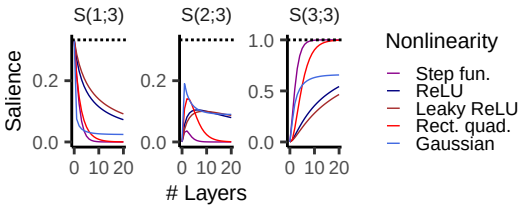

Figure 2: Representational salience (for three input components) in a random weight neural network with variable numbers of layers and nonlinearities.

Proposition 5.1. *For a random neural network with a (leaky) ReLU nonlinearity, as $L \to \infty$, $S(k; C) \to 0$ for $k < C$ and $S(C; C) \to 1$.*

We prove the proposition in Appendix B.5. It implies that very deep networks only encode the full conjunction. As a result, models effectively implement a look-up table, memorizing the training data without generalizing to novel combinations of components. Indeed, we will see below that the increasingly salient representation of the full conjunction will present significant challenges to compositional generalization long before it fully dominates this representation.

5.2 Kernel models suffer from memorization leak and shortcut bias

While there are many readout weights that can lead to minimal error, kernel models trained with gradient descent or ridge regression learn the weights with minimal ℓ_2-norm (see Section 2). We now characterize how this inductive bias influences compositional generalization. To characterize the full range of compositionally structured representations, we compute the behavior of the model directly using the kernel (see Appendix D). Importantly, our analysis clarifies the behavior of all random weight neural networks, including those covered in the previous section.

Symbolic addition suffers from a memorization leak. We first consider an example case ($S(1; 2) = 0.4, \mathcal{W} = \{0\}$). We find that though the model perfectly learns the training cases, it underestimates the test cases by a proportional factor (Fig. 3a). To understand why, we consider the model's functional form on the training cases: $f(x) = f_1(z_1) + f_2(z_2) + f_{12}(z)$ (see Section 4). Intuitively, ℓ_2-norm minimization tends to yield distributed weights. As a result, unless the salience of the conjunction $S(2; 2)$ is zero, the kernel model will associate some non-zero weight with it (i.e. $f_{12}(z_1, z_2) \neq 0$). This, in turn, necessarily distorts the generalization on the test set, $f(x) = f_1(z_1) + f_2(z_2)$.

We call this tendency to use the full conjunction (and its effect on generalization) "memorization leak." To characterize it more systematically (and add mathematical rigor to the intuition above), we analytically characterize model behavior on a general family of symbolic addition tasks:

Proposition 5.2. *Consider inputs $z_1, z_2 \in \{[v]\}_{v \in \mathcal{V}}, \mathcal{V} \subset \mathbb{R}$, with associated values v. We assume that the training set contains all pairs such that at least one component is $z_c \in \{[w]\}_{w \in \mathcal{W}}, \mathcal{W} \subset \mathcal{V}$ and that the average value in both $\mathcal{V}$ and $\mathcal{W}$ is zero. Then, model behavior on the test set is given by*

$$f([v_1], [v_2]) = m(v_1 + v_2), \quad m := \frac{p \cdot S(1;2)}{1 + (p-2)S(1;2)}, \quad p := |\mathcal{W}| \tag{4}$$

We prove the proposition in Appendix E.1. It implies that for this entire task family, model generalization is distorted by a proportional factor m (see Fig. 3b). The formula implies that $m < 1$ as long as $S(1; 2) < \frac{1}{2}$ (i.e. as long as $S(2; 2) > 0$). Further, m is smaller for lower $S(1; 2)$ and smaller training set size p. (Note that $\mathcal{W}$ corresponds to the rows/columns making up the example training sets in Fig. 1b.) Interestingly, these are the only two factors that influence m. In particular, while interpolation is often seen as easier than extrapolation, this does not impact model behavior here.

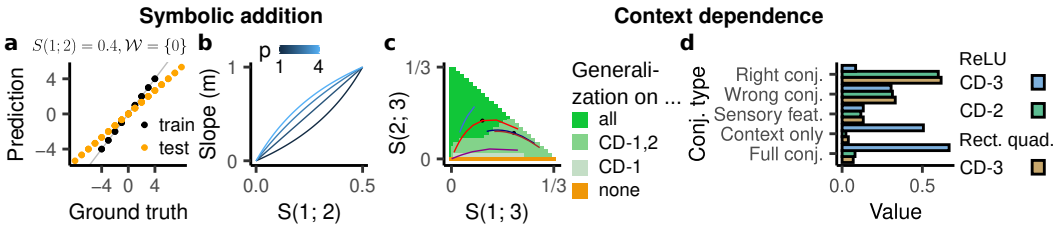

Figure 3: Kernel models' behavior on *(a,b)* symbolic addition and *(c,d)* context dependence. **a**, Model predictions on training and test set plotted against ground truth for an example case ($S(1;2) = 0.4$, $\mathcal{W} = \{0\}$). **b**, The slope of the test set as a function of $S(1;2)$ and the training set size $p = |\mathcal{W}|$. **c**, Generalization on context dependence as a function of representational salience. Trajectories of networks with different nonlinearities are highlighted (color scale see Fig. 2). **d**, Coefficients of the different conjunction types for two example networks with three layers and different nonlinearities.

More broadly, the memorization leak is a ubiquitous issue for statistical learning models trained on compositional tasks: ℓ_2-norm minimization tends to yield distributed weights and so if the conjunctive population is represented, the model will generally end up partially relying on this conjunction. As shown above, this necessarily distorts generalization. This issue also arises in tasks that involve directly decoding specific components, a popular task for evaluating disentangled representations (Locatello et al., 2019; Schott et al., 2022) (see also Appendix E.3 for a minimal example).

Context dependence suffers from a shortcut bias. Next, we empirically analyzed model generalization on context dependence across different representational geometries and training sets. We found that on a given task, each model either generalizes with 100% or 0% accuracy. For *CD-3* (the task version leaving out the largest subset of feature combinations), the model only generalizes when $S(2;3)$ is high relative to $S(1;3)$ (Fig. 3c, dark green area). As a result, whether the network generalizes is highly sensitive to the nonlinearity and depth of the network. In contrast, for *CD-2* and *CD-1*, a much wider range of representational geometries generalizes successfully.

To understand this phenomenon, we determined the total magnitude of model weights associated with the different conjunctions (Fig. 3d). We found that unsuccessful models (e.g. blue color in Fig. 3d) had much larger magnitudes associated with the context component and the full conjunction. Notably, on *CD-3*, context is highly correlated with the target and can predict 2/3 of the training data. Models with high $S(1;3)$ (context) and $S(3;3)$ (full conjunction) exploit the context shortcut and use the full conjunction to learn the remaining training data. This strategy explains why these models fail to generalize to the test set. For *CD-2*, context is much less predictive on the training data (accuracy of $\frac{9}{16}$). This explains why only very low $S(2;3)$ yields failure to generalize on *CD-2* or *CD-1*.

Our analysis shows how model and task structure interact to either give rise to a generalizable rule or a statistical shortcut (see Appendix E.3 for a minimal example). This highlights that for compositional tasks with strong spurious correlations, model behavior will be highly sensitive to architectural details affecting the representational geometry such as depth and nonlinearity. Indeed, this sensitivity to minor experimental details could explain why the literature has been so divided on the usefulness of disentangled representations.

6 Our theory can describe the behavior of deep networks on conjunction-wise additive tasks

So far, our analyses have been limited to simple kernel models, for their analytical tractability. We next test whether the insights developed from these simple models extend to a broader class of models: large-scale neural networks. We considered symbolic addition and context dependence on inputs created by concatenations of images from MNIST (Lecun et al., 1998) or CIFAR-10 (Krizhevsky et al., 2009) (see Appendix D). Each component corresponds to random samples from a particular category rather than the one-hot input considered so far and we study both in-distribution and compositional generalization. Notably, different image categories are not necessarily equally correlated with each other. To control for this, we randomly permuted the assignment of categories to components for each ($n = 10$) experiment. We considered several relevant vision architectures,

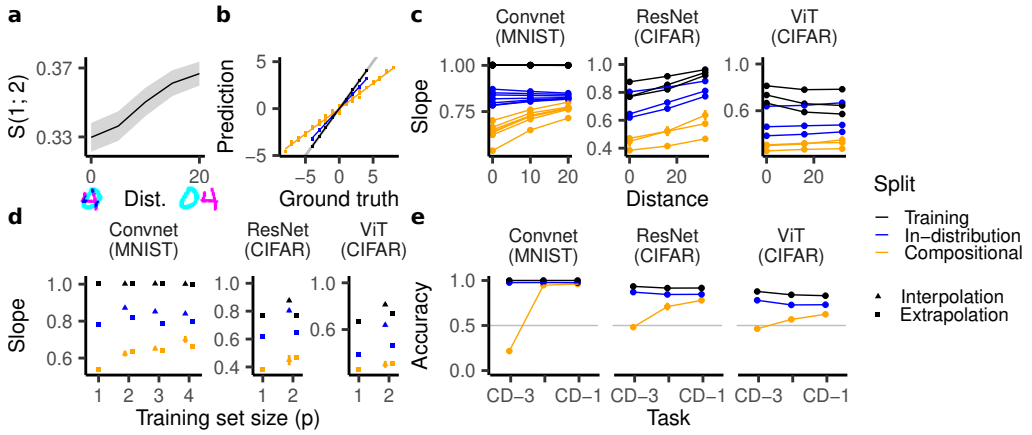

Figure 4: Testing our theory in deep networks trained on MNIST and CIFAR versions of compositional tasks. Ranges indicate mean $\pm$ one std. error (often too small to be visible). **a**, $S(1;2)$ in an intermediate ConvNet layer for different distances between digits. Lower distance yields a more conjunctive representation. **b**, Average model prediction for each combination of components plotted against the ground truth (MNIST, distance of zero, $\mathcal{W} = \{0\}$). Generalization on the compositional split is distorted by a proportional factor. **c**, **d**, Slope of this linear relationship across all datasets as a function of c, distance (each line corresponding to a particular $\mathcal{W}$) and d, training set (for a distance of zero). For MSE instead of slope, see Fig. 12. **e**, Accuracy on all variants of context dependence.

which we all trained with backpropagation: convolutional networks (ConvNets, LeCun et al., 1989) (trained on MNIST) and residual networks (ResNets, He et al., 2016) and Vision Transformers (ViTs, Dosovitskiy et al., 2020) (trained on CIFAR-10).

Spatial distance of components impact salience in internal representations. First, we examined how changes in the input structure impact the networks' representational geometry. We hypothesized that the ConvNets' local weight structure should produce a more conjunctive representation for digits that are closer together. To test this, we determined $S(1;2)$ in an intermediate layer of the network, averaging over different instances of all digits. We found that $S(1;2)$ was indeed smaller for lower distances (Fig. 4a), indicating a more conjunctive population. This suggests that varying the distance between two digits provides a practical way of manipulating $S(1;2)$; below we use this insight to test predictions about how conjunctivity influences generalization behavior.

Deep networks implement a conjunction-wise additive computation. Conjunction-wise additivity imposes a substantial computational restriction: on test set inputs, models can only add up values assigned to each conjunction of components seen during training. We therefore investigated whether the deep networks' computations could be captured in these terms. Remarkably, we found that such a model was highly predictive of the model responses (see Appendix D.2). This indicates that large-scale neural networks tend to implement conjunction-wise additive computations — at least when trained on conjunction-wise additive tasks. Our theory suggests that they do so because the statistical structure of the dataset makes such a computation natural.

Deep networks are impacted by a memorization leak on symbolic addition. Having found that our theory captures the networks' general computational structure, we investigated whether it was also able to explain their specific performance. We first considered symbolic addition, training the ConvNets on seven different training sets with 20,000 samples and the ResNets and ViTs on three training sets with 40,000 samples. We varied both the size of the training set and whether generalization required interpolation or also extrapolation (Appendix E.1.3).

We tested three theoretical predictions made by Proposition 5.2. First, our theory predicts that the model's test set response should be distorted by a proportional factor. To test this, we plotted the average model prediction for each combination of categories against the ground truth. We found that model predictions were indeed compressed relative to the ground truth, but still exhibited a strong linear relationship (Fig. 4b). This was also the case for ResNets and ViTs, which exhibited a slightly noisier linear relationship (Fig. 11). The memorization leak therefore affects generalization

in large-scale networks as well. This is especially significant as our theoretical argument suggests that memorization leaks likely arise for a broad range of compositional tasks.

We then estimated the slope of this linear dependency for each dataset. Our theory predicts that the compositional generalization slope should increase with increasing $S(1;2)$ (i.e. increasing distance between components, Fig. 4a). On ConvNets, we found that higher distance indeed increases the slope (Fig. 4c). This was not due to an increase in task difficulty, as in-distribution generalization is not systematically affected by distance. On ResNets and ViTs, higher distance also yielded a higher slope on the compositional generalization set (though the effect was much subtler for ViTs). (Notably, these networks did not perfectly predict the training split, which may affect the results.)

Finally, our theory predicts that larger training sets should increase the compositional generalization slope. Our experiments confirmed this prediction (Fig. 4d). Further, whether the training set required interpolation or also extrapolation did not systematically affect the resulting slope, as predicted by our theory (though interestingly, it did affect in-distribution generalization). However, there was one exception to this: for $p = 4$, the extrapolation dataset had a smaller slope than the interpolation data set. Our theory can therefore explain much but not all of the deep network behavior.

Deep networks are impacted by a shortcut bias on context dependence. Lastly, we trained the ConvNets on an MNIST version of context dependence using 30,000 training samples and the ResNets and ViTs on a CIFAR-10 version of the task using 40,000 training samples. Again, their behavior was aligned with the kernel theory's predictions, having better-than-chance accuracy on *CD-1* and *CD-2*, but having worse-than-chance accuracy on *CD-3* (Fig. 4e). This confirms that the deep networks also suffered from a context-driven shortcut on *CD-3*.

7 Discussion

Humans often generalize to new situations by stitching together concepts and knowledge from prior experience in new ways. Despite the broad importance of this ability (both for cognitive science and machine learning), a general theory of when and how neural networks accomplish compositional generalization has remained elusive. Here we have taken a step towards formalizing this relationship by characterizing kernel models with compositionally structured representations, a framework that captures neural network training in the kernel regime and more broadly lets us understand the impact of representational geometry on generalization. We found that they implement a specific compositional computation ("conjunction-wise additivity"). We then investigated how dataset statistics and representational geometry impact successful generalization, highlighting two failure modes arising from the inductive bias of gradient descent (memorization leak and shortcut bias). Finally, we validated our theory in deep neural networks trained on natural image data.

Our results show how simple statistical models can implement (or at least approximate) abstract rules like context dependence. However, they also highlight that building in compositional structure by itself is often insufficient for compositional generalization. Indeed, contextual generalization is highly sensitive to the specific representational geometry and training data. This may explain why investigations into compositional generalization and the benefits of disentangled representation have yielded such inconsistent results. Overall, our insights highlight the utility of kernel models in systematically investigating model generalization (or at least generating an initial hypothesis as to the important factors). We here used these models to characterize deep neural networks, but they could be equally useful for better understanding human compositional generalization.

While our work covers a broad range of different tasks, a number of limitations remain. We demonstrate that our theory captures many qualitative phenomena in deep neural networks, but do not provide any quantitative bounds. Further, though out of scope here, it would be interesting to test this theory in extremely large-scale models, e.g. large language models. Future work should also consider inputs and outputs with broader ranges of formats (e.g. sequences with variable length) as well as representations that are not compositionally structured. Most importantly, our theory is limited to a particular learning mechanism (fixed features with linear readout); other learning mechanisms could either implement a conjunction-wise additive model that overcomes the failure modes highlighted here or implement novel compositional computations beyond conjunction-wise additivity (we give an initial example of this in Appendix F). By clarifying the general relationship between dataset statistics and compositional generalization, our theory provides an important foundation for such advances.

REFERENCES

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, and Kevin Rizk. Generalization on the Unseen, Logic Reasoning and Degree Curriculum. June 2023. URL https://openreview.net/forum?id=3dqwXb1te4.

Mustafa Abdool, Andrew J Nam, and James L McClelland. Continual learning and out of distribution generalization in a systematic reasoning task. In *MATH-AI: The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS*, volume 23, 2023.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural Module Networks, July 2017. URL http://arxiv.org/abs/1511.02799. arXiv:1511.02799 [cs].

Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. Systematic Generalization: What Is Required and Can It Be Learned?, April 2019. URL http://arxiv.org/abs/1811.12889. arXiv:1811.12889 [cs].

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, October 2018. URL http://arxiv.org/abs/1806.01261. arXiv:1806.01261 [cs, stat].

Mark E. Bouton, James B. Nelson, and Juan M. Rosas. Stimulus generalization, context change, and forgetting. *Psychological Bulletin*, 125(2):171–186, 1999. ISSN 1939-1455. doi: 10.1037/0033-2909.125.2.171. Place: US Publisher: American Psychological Association.

Alon Brutzkus and Amir Globerson. Why do Larger Models Generalize Better? A Theoretical Perspective via the XOR Problem. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 822–830. PMLR, May 2019. URL https://proceedings.mlr.press/v97/brutzkus19b.html. ISSN: 2640-3498.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Out-of-Distribution Generalization in Kernel Regression. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12600–12612. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/691dcb1d65f31967a874d18383b9da75-Abstract.html.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, May 2021b. ISSN 2041-1723. doi: 10.1038/s41467-021-23103-1. URL https://www.nature.com/articles/s41467-021-23103-1. Publisher: Nature Publishing Group.

Abdulkadir Canatar, Jenelle Feather, Albert Wakhloo, and SueYeon Chung. A Spectral Theory of Neural Prediction and Alignment. November 2023. URL https://openreview.net/forum?id=5B1ZK60jWn.

Stephanie C. Y. Chan, Ishita Dasgupta, Junkyung Kim, Dharshan Kumaran, Andrew K. Lampinen, and Felix Hill. Transformers generalize differently from information stored in context vs in weights, October 2022. URL http://arxiv.org/abs/2210.05675. arXiv:2210.05675 [cs].

Lénaïc Chizat and Francis Bach. Implicit Bias of Gradient Descent for Wide Two-layer Neural Networks Trained with the Logistic Loss. In *Proceedings of Thirty Third Conference on Learning Theory*, pp. 1305–1338. PMLR, July 2020. URL https://proceedings.mlr.press/v125/chizat20a.html. ISSN: 2640-3498.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On Lazy Training in Differentiable Programming. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/ae614c557843b1df326cb29c57225459-Abstract.html.

Youngmin Cho and Lawrence Saul. Kernel Methods for Deep Learning. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://papers.nips.cc/paper/2009/hash/5751ec3e9a4feab575962e78e006250d-Abstract.html.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, December 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL https://doi.org/10.1007/BF02551274.

Ishita Dasgupta, Erin Grant, and Tom Griffiths. Distinguishing rule and exemplar-based generalization in learning systems. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 4816–4830. PMLR, June 2022. URL https://proceedings.mlr.press/v162/dasgupta22b.html. ISSN: 2640-3498.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and others. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Aparna Elangovan, Jiayuan He, and Karin Verspoor. Memorization vs. Generalization: Quantifying Data Leakage in NLP Performance Evaluation, February 2021. URL http://arxiv.org/abs/2102.01818. arXiv:2102.01818 [cs].

Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, N. Siddharth, Brooks Paige, Dana H. Brooks, Jennifer Dy, and Jan-Willem Meent. Structured Disentangled Representations. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 2525–2534. PMLR, April 2019. URL https://proceedings.mlr.press/v89/esmaeili19a.html. ISSN: 2640-3498.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pp. 954–959, New York, NY, USA, June 2020. Association for Computing Machinery. ISBN 978-1-4503-6979-4. doi: 10.1145/3357713.3384290. URL https://dl.acm.org/doi/10.1145/3357713.3384290.

Jerry A. Fodor and Zenon W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, March 1988. ISSN 0010-0277. doi: 10.1016/0010-0277(88)90031-5. URL https://www.sciencedirect.com/science/article/pii/0010027788900315.

Steven M. Frankland and Joshua D. Greene. Concepts and Compositionality: In Search of the Brain's Language of Thought. *Annual Review of Psychology*, 71(1):273–303, 2020. doi: 10.1146/annurev-psych-122216-011829. URL https://doi.org/10.1146/annurev-psych-122216-011829. _eprint: https://doi.org/10.1146/annurev-psych-122216-011829.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL https://www.nature.com/articles/s42256-020-00257-z. Number: 11 Publisher: Nature Publishing Group.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Graeme S. Halford, William H. Wilson, and Steven Phillips. Relational knowledge: the foundation of higher cognition. *Trends in Cognitive Sciences*, 14(11):497–505, November 2010. ISSN 1364-6613. doi: 10.1016/j.tics.2010.08.005. URL https://www.sciencedirect.com/science/article/pii/S1364661310002020.

Insu Han, Amir Zandieh, Jaehoon Lee, Roman Novak, Lechao Xiao, and Amin Karbasi. Fast Neural Kernel Embeddings for General Activations, September 2022. URL http://arxiv.org/abs/2209.04121. arXiv:2209.04121 [cs, stat].

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. pp. 1026–1034, 2015. URL https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming Auto-Encoders. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2011*, pp. 44–51, Berlin, Heidelberg, 2011. Springer. ISBN 978-3-642-21735-7. doi: 10.1007/978-3-642-21735-7_6.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, January 1989. ISSN 0893-6080. doi: 10.1016/0893-6080(89)90020-8. URL https://www.sciencedirect.com/science/article/pii/0893608089900208.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. Compositionality Decomposed: How do Neural Networks Generalise? *Journal of Artificial Intelligence Research*, 67:757–795, April 2020. ISSN 1076-9757. doi: 10.1613/jair.1.11674. URL https://www.jair.org/index.php/jair/article/view/11674.

Takuya Ito, Tim Klinger, Douglas H. Schultz, John D. Murray, Michael W. Cole, and Mattia Rigotti. Compositional generalization through abstract representations in human and artificial neural networks, September 2022. URL http://arxiv.org/abs/2209.07431. arXiv:2209.07431 [q-bio].

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Devon Jarvis, Richard Klein, Benjamin Rosman, and Andrew M. Saxe. On The Specialization of Neural Modules. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Fh97BDaR6I.

Ziwei Ji, Miroslav Dudík, Robert E Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses. In *Conference on Learning Theory*, pp. 2109–2136. PMLR, 2020.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. pp. 2901–2910, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Johnson_CLEVR_A_Diagnostic_CVPR_2017_paper.html.

Frank Jäkel, Bernhard Schölkopf, and Felix A Wichmann. Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin & Review*, 15: 256–271, 2008. Publisher: Springer.

Frank Jäkel, Bernhard Schölkopf, and Felix A. Wichmann. Does Cognitive Science Need Kernels? *Trends in Cognitive Sciences*, 13(9):381–388, September 2009. ISSN 1364-6613, 1879-307X. doi: 10.1016/j.tics.2009.06.002. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(09)00143-0. Publisher: Elsevier.

Alex Krizhevsky, Geoffrey Hinton, and others. Learning multiple layers of features from tiny images. 2009. Publisher: Toronto, ON, Canada.

Sebastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive Decoders for Latent Variables Identification and Cartesian-Product Extrapolation. November 2023. URL https://openreview.net/forum?id=R6KJN1AUAR.

Brenden Lake and Marco Baroni. Generalization without systematicity: 35th International Conference on Machine Learning, ICML 2018. *35th International Conference on Machine Learning, ICML 2018*, pp. 4487–4499, 2018. URL http://www.scopus.com/inward/record.url?scp=85057241154&partnerID=8YFLogxK. Publisher: International Machine Learning Society (IMLS).

Brenden M. Lake and Marco Baroni. Human-like systematic generalization through a meta-learning neural network. *Nature*, 623(7985):115–121, November 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06668-3. URL https://www.nature.com/articles/s41586-023-06668-3. Publisher: Nature Publishing Group.

Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, December 2015. doi: 10.1126/science.aab3050. URL https://www.science.org/doi/full/10.1126/science.aab3050. Publisher: American Association for the Advancement of Science.

Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40:e253, January 2017. ISSN 0140-525X, 1469-1825. doi: 10.1017/S0140525X16001837. URL https://www.cambridge.org/core/journals/behavioral-and-brain-sciences/article/building-machines-that-learn-and-think-like-people/A9535B1D745A0377E16C590E14B94993. Publisher: Cambridge University Press.

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Martha Lewis, Nihal V. Nayak, Peilin Yu, Qinan Yu, Jack Merullo, Stephen H. Bach, and Ellie Pavlick. Does CLIP Bind Concepts? Probing Compositionality in Large Image Models, March 2023. URL http://arxiv.org/abs/2212.10537. arXiv:2212.10537 [cs].

Samuel Lippl, Kenneth Kay, Greg Jensen, Vincent P. Ferrera, and L. F. Abbott. A mathematical theory of relational generalization in transitive inference. *Proceedings of the National Academy of Sciences*, 121(28):e2314511121, July 2024. doi: 10.1073/pnas.2314511121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2314511121. Publisher: Proceedings of the National Academy of Sciences.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Elena Lorenzi, Matilde Perrino, and Giorgio Vallortigara. Numerosities and Other Magnitudes in the Brains: A Comparative View. *Frontiers in Psychology*, 12, April 2021. ISSN 1664-1078. doi: 10.3389/fpsyg.2021.641994. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2021.641994/full. Publisher: Frontiers.

Kaifeng Lyu and Jian Li. Gradient Descent Maximizes the Margin of Homogeneous Neural Networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJeLIgBKPS.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A Kernel-Based View of Language Model Fine-Tuning. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 23610–23641. PMLR, July 2023. URL https://proceedings.mlr.press/v202/malladi23a.html. ISSN: 2640-3498.

Brendan O. McGonigle and Margaret Chalmers. Are monkeys logical? *Nature*, 267:694–696, 1977. ISSN 1476-4687. doi: 10.1038/267694a0. Place: United Kingdom Publisher: Nature Publishing Group.

Eric Mitchell, Chelsea Finn, and Chris Manning. Challenges of acquiring compositional inductive biases via meta-learning. In *AAAI Workshop on Meta-Learning and MetaDL Challenge*, pp. 138–148. PMLR, 2021.

Sarthak Mittal, Yoshua Bengio, and Guillaume Lajoie. Is a Modular Architecture Enough? *Advances in Neural Information Processing Systems*, 35:28747–28760, December 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b8d1d741f137d9b6ac4f3c1683791e4a-Abstract-Conference.html.

Vaishnavh Nagarajan, Anders Andreassen, and Behnam Neyshabur. Understanding the Failure Modes of Out-of-Distribution Generalization, April 2021. URL http://arxiv.org/abs/2010.15775. arXiv:2010.15775 [cs, stat].

Jeanne E Parker and Debra L Hollister. The Cognitive Science Basis for Context. In Patrick Brézillon and Avelino J. Gonzalez (eds.), *Context in Computing: A Cross-Disciplinary Approach for Modeling the Real World*, pp. 205–219. Springer, New York, NY, 2014. ISBN 978-1-4939-1887-4. doi: 10.1007/978-1-4939-1887-4_14. URL https://doi.org/10.1007/978-1-4939-1887-4_14.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and others. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Yi Ren, Samuel Lavoie, Michael Galkin, Danica J. Sutherland, and Aaron C Courville. Improving Compositional Generalization using Iterated Learning and Simplicial Embeddings. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 60547–60572. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/be7430d22a4dae8516894e32f2fcc6db-Paper-Conference.pdf.

Mattia Rigotti, Omri Barak, Melissa R Warden, Xiao-Jing Wang, Nathaniel D Daw, Earl K Miller, and Stefano Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497 (7451):585–590, 2013. Publisher: Nature Publishing Group UK London.

Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In Alina Beygelzimer and Daniel Hsu (eds.), *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pp. 2667–2690. PMLR, June 2019. URL https://proceedings.mlr.press/v99/savarese19a.html.

Andrew Saxe, Shagun Sodhani, and Sam Jay Lewallen. The Neural Race Reduction: Dynamics of Abstraction in Gated Networks. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 19287–19309. PMLR, June 2022. URL https://proceedings.mlr.press/v162/saxe22a.html. ISSN: 2640-3498.

Margaret L Schlichting and Alison R Preston. Memory integration: neural mechanisms and implications for behavior. *Current opinion in behavioral sciences*, 1:1–8, 2015. Publisher: Elsevier.

Lukas Schott, Julius Von Kügelgen, Frederik Träuble, Peter Vincent Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual Representation Learning Does Not Generalize Strongly Within the Same Domain. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9RUHPlladgh.

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wołczyk, Alexandra Proca, Johannes von Oswald, Razvan Pascanu, João Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally, March 2024. URL http://arxiv.org/abs/2312.15001. arXiv:2312.15001 [cs].

Philipp Schwartenbeck, Alon Baram, Yunzhe Liu, Shirley Mark, Timothy Muller, Raymond Dolan, Matthew Botvinick, Zeb Kurth-Nelson, and Timothy Behrens. Generative replay underlies compositional inference in the hippocampal-prefrontal circuit. *Cell*, October 2023. ISSN 0092-8674. URL https://doi.org/10.1016/j.cell.2023.09.004. Place: United States.

Bernhard Schölkopf. The Kernel Trick for Distances. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://papers.nips.cc/paper_files/paper/2000/hash/4e87337f366f72daa424dae11df0538c-Abstract.html.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The Pitfalls of Simplicity Bias in Neural Networks, October 2020. URL http://arxiv.org/abs/2006.07710. arXiv:2006.07710 [cs, stat].

Hannah Sheahan, Fabrice Luyckx, Stephanie Nelli, Clemens Teupe, and Christopher Summerfield. Neural state space alignment for magnitude generalization in humans and recurrent networks. *Neuron*, 109(7):1214–1226.e8, April 2021. ISSN 0896-6273. doi: 10.1016/j.neuron.2021.02.004. URL https://www.sciencedirect.com/science/article/pii/S0896627321000787.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1): 2822–2878, 2018. Publisher: JMLR. org.

Kelsey N Spalding, Margaret L Schlichting, Dagmar Zeithamova, Alison R Preston, Daniel Tranel, Melissa C Duff, and David E Warren. Ventromedial prefrontal cortex is necessary for normal associative inference and memory integration. *Journal of Neuroscience*, 38(15):3767–3775, 2018. Publisher: Soc Neuroscience.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola,

Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima, Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models, June 2023. URL http://arxiv.org/abs/2206.04615. arXiv:2206.04615 [cs, stat].

Jordan A. Taylor and Richard B. Ivry. Context-dependent generalization. *Frontiers in Human Neuroscience*, 7, May 2013. ISSN 1662-5161. doi: 10.3389/fnhum.2013.00171. URL https://www.frontiersin.org/articles/10.3389/fnhum.2013.00171. Publisher: Frontiers.

Frederik Träuble, Elliot Creager, Niki Kilbertus, Francesco Locatello, Andrea Dittadi, Anirudh Goyal, Bernhard Schölkopf, and Stefan Bauer. On Disentangled Representations Learned from Correlated Data. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10401–10412. PMLR, July 2021. URL https://proceedings.mlr.press/v139/trauble21a.html. ISSN: 2640-3498.

Russell Tsuchida, Farbod Roosta-Khorasani, and Marcus Gallagher. Invariance of Weight Distributions in Rectified MLPs, May 2018. URL http://arxiv.org/abs/1711.09090. arXiv:1711.09090 [cs, stat].

Russell Tsuchida, Fred Roosta, and Marcus Gallagher. Richer priors for infinitely wide multi-layer perceptrons, November 2019. URL http://arxiv.org/abs/1911.12927. arXiv:1911.12927 [cs, stat].

Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are Disentangled Representations Helpful for Abstract Visual Reasoning? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/bc3c4a6331a8a9950945a1aa8c95ab8a-Abstract.html.

Gal Vardi and Ohad Shamir. Implicit Regularization in ReLU Networks with the Square Loss. In *Proceedings of Thirty Fourth Conference on Learning Theory*, pp. 4224–4258. PMLR, July 2021. URL https://proceedings.mlr.press/v134/vardi21b.html. ISSN: 2640-3498.

Colin G. West. Advances in apparent conceptual physics reasoning in GPT-4, March 2023. URL https://ui.adsabs.harvard.edu/abs/2023arXiv230317012W. Publication Title: arXiv e-prints ADS Bibcode: 2023arXiv230317012W.

James C. R. Whittington, Will Dorrell, Surya Ganguli, and Timothy Behrens. Disentanglement with Biological Constraints: A Theory of Functional Cell Types. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9Z_GfhZnGH.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable Compositional Generalization for Object-Centric Learning. *arXiv preprint arXiv:2310.05327*, 2023a.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional Generalization from First Principles. *Advances in Neural Information Processing Systems*, 36:6941–6960, December 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/15f6a10899f557ce53fe39939af6f930-Abstract-Conference.html.

Bin Wu, Jinyuan Fang, Xiangxiang Zeng, Shangsong Liang, and Qiang Zhang. Adaptive compositional continual meta-learning. In *International Conference on Machine Learning*, pp. 37358–37378. PMLR, 2023.

Zhiwei Xu, Yutong Wang, Spencer Frei, Gal Vardi, and Wei Hu. Benign Overfitting and Grokking in ReLU Networks for XOR Cluster Data, October 2023. URL http://arxiv.org/abs/2310.02541. arXiv:2310.02541 [cs, stat].

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Michael C. Mozer, and Yoram Singer. Identity Crisis: Memorization and Generalization under Extreme Overparameterization, January 2020. URL http://arxiv.org/abs/1902.04698. arXiv:1902.04698 [cs, stat].

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. Publisher: ACM New York, NY, USA.

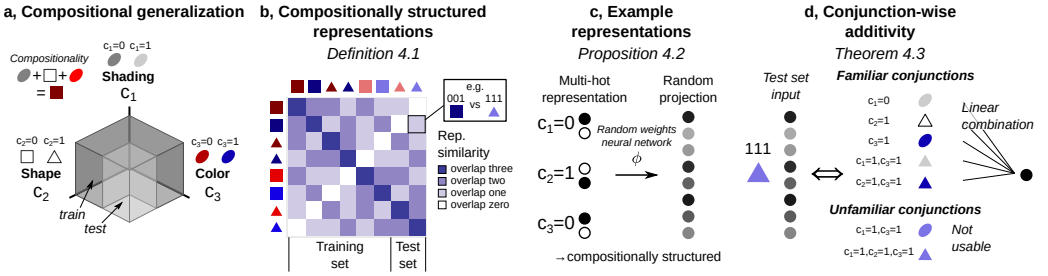

Figure 5: Schematic overview of our findings in Section 4. **a**, We consider compositional datasets consisting of different categorical components, in this example dark vs. light shading, shape, and color. **b**, We assume compositionally structured representations for which trials with the same number of overlaps have identical similarities. **c**, We find that a random weights neural network conserves compositional structure, i.e. if its input is compositionally structured then so is its output. **d**, We find that under this condition, linear readout models trained with gradient descent are constrained to implementing a conjunction-wise additive computation. This means that they add up values for each conjunction they have seen during training. In our example, for instance, they have previously seen inputs with $l_1 = 1$ and $l_2 = 1$, but they have not seen inputs with $l_1 = 1$ and $l_3 = 1$.

A Mathematical analysis

A.1 Generalized compositionally structured representations

In the main text, we considered "compositionally structured" representations. This is the class of representations $\phi(z)$ whose kernel $K_\phi(z, z') = \phi(z)^T \phi(z')$ only depends on the number of components z and z' have in common. Here we define a slightly more general class of representations whose kernel depends on the *identity* of the components they have in common, i.e.

$$O(z, z') := \{c = 1, \ldots, C | z_c = z'_c\}. \tag{5}$$

Thus, pairs of representations overlapping in the first component may have a different similarity than pairs of representations overlapping in the second component, but any pair of representations overlapping in the first component still need to have the same similarity. This case captures, for example, cases where one feature is more salient than another and therefore influences the representational similarity more strongly, or where conjunctions between certain components are more saliently represented than conjunctions between other components. For example, perhaps our input consists of two objects with different shapes and colors. In that case, we may wish to represent the conjunction between shape and color of the first object more strongly than the conjunction between the shape of the first object and the color of the second.

Note that we consider the constrained definition of compositionally structured representations in the main text purely for didactical purposes. In particular, our observations in Appendices A.3 and A.4 extend to all generalized compositionally structured representations.

A.2 Why are we considering compositionally structured representations?

So far, we have motivated the concept of compositionally structured representations in terms of the fact that disentangled representations as well as many nonlinear transforms of disentangled representations are compositionally structured. Here we discuss a more conceptual motivation for them: compositionally structured representations guarantee that the only basis for generalization is the compositional nature of the data. This is because we ensure that the model has no knowledge about certain components that may be more similar to each other. Otherwise, the component-wise similarity could serve as a basis for generalization. For example, in transitive equivalence (Fig. 1d), if items that are in the same equivalence class are represented as more similar to each other, that would serve as a basis for generalization on the task that is not compositional in nature. For this reason, we are particularly interested in compositionally structured representations, as they ensure that kernel models are only able to generalize compositionally.

Characterizing compositional generalization in non-compositionally structured representations requires us to meaningfully distinguish between compositional and non-compositional aspects of their generalization. We present an example of such an analysis in Appendix A.5, proving that if the non-compositional representational components are random, the models still generalize in a conjunction-wise additive manner in expectation.

A.3 Nonlinear transformations and compositional structure

A broad range of transforms ϕ induces a kernel $K_\phi(z, z') = \phi(z)^T\phi(z')$ that only depends on the input similarity $K(z, z') = z^T z'$. In particular (as noted in the main text), this condition is satisfied by the hidden layers and neural tangent kernel of randomly initialized neural networks (in the infinite-width limit) (Han et al., 2022):

Definition A.1. Given an input $z \in \mathbb{R}^d$, a network depth $L \geq 2$, and a set of widths $H_1, \ldots, H_L \in \mathbb{N}$, we define a neural network by recursively defining the operation $\phi^{(l)}$ of the l^{th} layer as

$$a^{(0)} := z, \quad a^{(l)} := \phi^{(l)}(a^{(l-1)}) := \sigma\left(\tfrac{1}{\sqrt{H_l}}\left(W^{(l)}a^{(l-1)} + b^{(l)}\right)\right), \quad W^{(l)} \in \mathbb{R}^{H_l \times H_{l-1}} \tag{6}$$

where $W^{(l)}$ and $b^{(l)}$ are i.i.d. sampled from a random distribution and $\sigma : \mathbb{R} \to \mathbb{R}$ is a nonlinearity. The complete transform is then given by $\phi := \phi^{(L)} \circ \cdots \circ \phi^{(1)}$.

The infinite-width limit is characterized by $H_1, \ldots, H_{L-1} \to \infty$ (the order of these limits does not matter). This setup includes, in particular, the random feature model investigated by Abbe et al. (2023).

Further, the condition is met by most commonly used kernel functions including any radial basis function that depends on the Euclidean ℓ_2-norm (e.g. the Gaussian kernel). Any nonlinear transform that meets this condition also conserves compositional structure, i.e. if z is compositionally structured then so is $\phi(z)$. This allows us to derive computational restrictions on this broad range of models.

A.4 Proof of Theorem 4.2

Theorem 4.2. *For any kernel model f with a compositionally structured representation, we can find conjunction-wise functions $f_J : \prod_{c\in J} Z_c \to \mathbb{R}$, where $J \subseteq \{1, \ldots, C\}$, such that for any input $x \in \mathbb{R}^d$ representing components $z \in Z$, the model response is given by*

$$f(x) = \textstyle\sum_{J\in \mathit{Conj}(z|Z^{\mathit{train}})} f_J(z_J), \quad z_J := (z_c)_{c\in J}. \tag{3}$$

Proof. Because the kernel K is compositionally structured, its similarity $K(z, z')$ only depends on the overlap $O(z, z') \subseteq \{1, \ldots, C\}$ (see definition in Eq. (5)). We denote the similarity for inputs overlapping in $S \subseteq \{1, \ldots, C\}$ by κ_S and define set of training items overlapping with $z \in Z^{\text{test}}$ in S as

$$Z^{\text{train}}(z, J) := \left\{z^{\text{tr}} \in Z^{\text{train}} | \forall_{c\in J} z_c = z_c^{\text{tr}}\right\}. \tag{7}$$

The key idea is to decompose Z^{train} into these different overlaps in order to separate the sum into its components. However, by our definition, the datasets $Z^{\text{train}}(z, J)$ are not disjoint. Indeed, $S \subseteq S'$ implies $Z^{\text{train}}(z, J) \subseteq Z^{\text{train}}(z, J')$ and in particular $Z^{\text{train}}(z, \emptyset) = Z^{\text{train}}$. To adjust for this, we define δ_J as the similarity added by κ_S to the similarity between conjunctions with one component fewer, recursively defining

$$\delta_\emptyset = \kappa_\emptyset, \quad \delta_J = \kappa_J - \sum_{J'\subsetneq J} \delta_{J'}. \tag{8}$$

We then decompose

$$f(z) = \sum_{z^{\text{tr}}\in Z^{\text{train}}} a_{z^{\text{tr}}} K(z, z^{\text{tr}}) = \sum_{J\subseteq\{1,\ldots,C\}} \delta_S \sum_{z^{\text{tr}}\in Z^{\text{train}}(z,J)} a_{z^{\text{tr}}}. \tag{9}$$

This equality obtains because for each $z \in Z, z^{\text{tr}} \in Z^{\text{train}}$,

$$\sum_{J: z^{\text{tr}}\in Z^{\text{train}}(z,J)} \delta_J = \delta_{O(z,z^{\text{tr}})} + \sum_{J'\subsetneq O(z,z^{\text{tr}})} \delta_{J'} = \kappa_{O(z,z^{\text{tr}})} = K(z, z^{\text{tr}}), \tag{10}$$

which is true by definition. We note that for $J \notin \text{Conj}(z|Z^{\text{train}})$, $Z^{\text{train}}(z, J) = \emptyset$. Defining

$$f_J(z) := \delta_J \sum_{z^{\text{tr}} \in Z^{\text{train}}(z,J)} a_{z^{\text{tr}}}, \tag{11}$$

proves the proposition. □

A.5 Non-compositionally structured representations

So far, we have considered representations that are compositionally structured. In practice, however, representations will almost never be exactly compositionally structured. Even when there is no specific similarity structure within components, there will be random noise in the representations. Here we characterize one such scenario, demonstrating that in expectation, these representations still yield conjunction-wise additive computations.

Proposition A.2. *Consider an input with C components. We consider a representation that represents each component and conjunction of components z_J, $J \subseteq \{1, \dots, C\}$ by a random vector $x_J[z_J] \in \mathbb{R}^d$, $x_J \sim \mathcal{N}(0, \sigma_k^2)$, $\sigma_k^2 > 0$, where $k = |J|$. The representation itself is given by the sum of all these vectors: $x = \sum_{J \subseteq \{1,\dots,C\}} x_J[z_J]$. All $x_J[z_J]$ are sampled independently from each other. This means that the similarity between different components as well as the entanglement of different components varies randomly. As a result, the kernel regression estimator $f(z)$ is not conjunction-wise additive. However, its expectation, $\mathbb{E}[f(z)]$, is conjunction-wise additive, indicating that all deviations from conjunction-wise additivity arise from random noise and cannot be used for systematic generalization.*

Proof. We consider the training representation $X = (x[z_J])_{z \in L^{\text{train}}}$, $X \in \mathbb{R}^{N_{\text{train}} \times d}$ and the test representation $\tilde{X} = (x[z_J])_{z \in L^{\text{test}}}$, $\tilde{X} \in \mathbb{R}^{N_{\text{test}} \times d}$. This gives rise to the training kernel $K = XX^T$ and the train-test kernel, $\tilde{K} = \tilde{X}X^T$. The dual coefficients are given by $a = K^{-1}y$. We now consider a test set input $\tilde{x} = \tilde{x}[\tilde{z}]$ representing the underlying components $\tilde{z}$. Denoting the similarity to the training set by $k(\tilde{x}, X) = X\tilde{x} \in \mathbb{R}^{N_{\text{train}}}$, the test set behavior is given by

$$f(\tilde{x}) = a^T k(\tilde{x}, X) = \sum_{J \in \text{Conj}(\tilde{z}|Z^{\text{train}})} a^T X x_J[z_J] + \sum_{J \notin \text{Conj}(\tilde{z}|Z^{\text{train}})} a^T X x_J[z_J] =: f_1(\tilde{x}) + f_2(\tilde{x}), \tag{12}$$

where we've simply split up the sum into the conjunction-wise additive part and the remainder. Clearly, this remainder, $f_2(\tilde{x}) := \sum_{J \notin \text{Conj}(\tilde{z}|Z^{\text{train}})} a^T X x_J[z_J]$, will generally not be zero. However, we will now prove that $\mathbb{E}[f_2(\tilde{x})] = 0$. As $f_1(\tilde{x})$ is conjunction-wise additive, this will prove the proposition. To do so, we note that we can assume that we have first sampled all Gaussian vectors relevant for the training set (we will denote this set of random variables by $\mathcal{X}^{\text{train}}$) and subsequently sample the set of Gaussian vectors only relevant for the test trial, i.e. $(\tilde{x}_J)_{J \notin \text{Conj}(\tilde{z}|Z^{\text{train}})}$ (we will denote this set of random variables by $\mathcal{X}^{\text{test}}$). Then,

$$\mathbb{E}[f_2(\tilde{x})] = \mathbb{E}_{\mathcal{X}^{\text{train}}}\left[\mathbb{E}_{\mathcal{X}^{\text{test}}}\left[f_2(\tilde{x})|\mathcal{X}^{\text{train}}\right]\right] = \sum_{J \notin \text{Conj}(\tilde{z}|Z^{\text{train}})} \mathbb{E}_{\mathcal{X}^{\text{train}}}\left[a^T X \mathbb{E}_{\mathcal{Z}^{\text{test}}}\left[x_J[\tilde{z}_J]|\mathcal{Z}^{\text{train}}\right]\right]. \tag{13}$$

This is zero, as $\mathbb{E}_{\mathcal{X}^{\text{test}}}\left[x_J[z_J]\right] = \mathbb{E}_{\mathcal{X}^{\text{test}}}\left[x_J[z_J]|\mathcal{X}^{\text{train}}\right] = 0$, which follows from the fact that $x_J[z_J]$ for $J \notin \text{Conj}(\tilde{z}|Z^{\text{train}})$ is sampled independently from $\mathcal{X}^{\text{train}}$. □

To test our theory empirically, we sample a range of Gaussian representations and train them on symbolic addition, transitive equivalence, and context dependence. For symbolic addition and transitive equivalence, we consider representations with expected $S(1; 2) \in [0.1, 1]$ using ten equally spaced values (i.e. $0.1, 0.2, \dots$). For context dependence, we consider two types of representation: rep. 1, where $\sigma_1 = 1, \sigma_2 = 0.5, \sigma_3 = 0.1$, (i.e. single components are most saliently represented), and rep. 2, where $\sigma_1 = 0.5, \sigma_2 = 1, \sigma_3 = 0.1$ (i.e. conjunctions of two components are most saliently represented). We consider representations with $d = 100$ and sample 100 instances of each representations. We then fit each representation to these three tasks.

We first estimate these models' additivity on symbolic addition and context dependence. We find that while many model instance are highly additive, some can be highly non-additive (Fig. 6a). However, when averaging across the model behavior 100 randomly sampled representations, this average

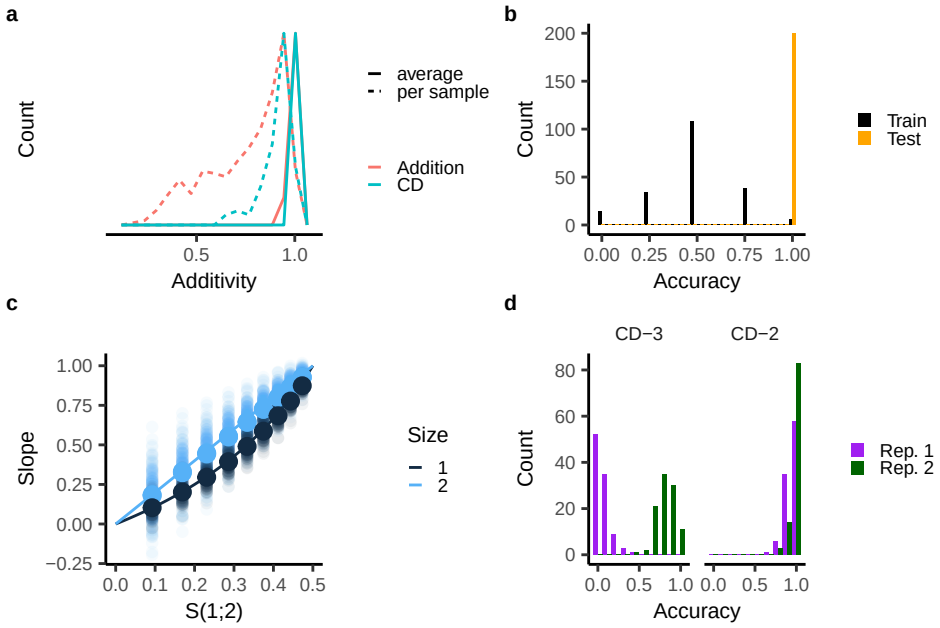

Figure 6: Analysis of randomly sampled representations as considered in Proposition A.2. In all cases we sample 100-dimensional representations. For addition, we consider representations with weights that explore $S(1;2) \in [0.1, 1]$ and for context dependence, we consider representations where $(\sigma_1, \sigma_2, \sigma_3) \in \{(0.5, 1, 0.1), (1, 0.5, 0.1)\}$, where σ_k indicates the standard deviation of the Gaussian vector representing conjunctions of k components. **a**, We compare the additivity of individual models to the additivity of the model behavior averaged across 100 randomly sampled representations. While individual models may be severely non-additive, their average behavior is consistently highly additive, empirically confirming our proposition. **b**, As a result, when these models are trained on transitive equivalence, they also systematically exhibit chance performance, as they are still, on average, constrained to be conjunction-wise additive. **c**, We then estimated the slope of the model trained on symbolic addition for each random instance of a representation against corresponding representation's $S(1;2)$. The individual slope estimates are depicted by the small translucent dots, whereas the average across all random instances is depicted by the larger points. Finally, our theoretical predictions in the exactly compositionally structured case are depicted by the lines. The average model behaviors are well described by our theory. **d**, We plot the distribution of generalization accuracies on context dependence across different model seeds, the two considered representations, and for CD-3 and CD-2. On CD-3, the representation with a higher salience for individual components (rep. 1, $(\sigma_1, \sigma_2, \sigma_3) = (0.5, 1, 0.1)$ performs systematically below chance, whereas the representation with a higher salience for conjunctions of two components (rep. 2, $(\sigma_1, \sigma_2, \sigma_3) = (1, 0.5, 0.1)$) generalizes above chance. In contrast, on CD-2, both representation exhibit better-than-chance-accuracy.

behavior is perfectly described by a conjunction-wise additive function. This empirically confirms our proposition.

Our proposition implies that these non-compositionally structured representations can still only systematically generalize on conjunction-wise additive tasks. To confirm this insight, we plotted the distribution of accuracies on the transitive equivalence problem across all random model instances. We found that while the models consistently learned the training set, they were indeed unable to generalize to the test set (Fig. 6b).

Finally, we investigated whether our analysis in Section 5 predicted the behavior of randomly sampled representations. Importantly, we have no theoretical guarantees for this scenario. Interestingly, however, the model behaviors were still well predicted by our theory. Specifically, we estimated the slopes that best described the compositional generalization behavior on symbolic addition for different training sets (as in Fig. 4). We found that across different model instances, these slopes were highly varied (Fig. 6c). However, the average slope across all random samples was well predicted by our analytical theory.

Similarly, our theoretical insights on context dependence provided meaningful insight into the generalization behavior of these randomly sampled models. Specifically, on CD-3, the model more saliently representing the single components failed to generalize, whereas the model more saliently representing conjunctions of two components generalized above chance. In contrast, on CD-2, both models generalized above chance. This indicates that our insights on how prevalent shortcut biases are for different training datasets extends to randomly sampled representations.

B Representational salience: a metric for compositionally structured representations

B.1 Definition

Below we formally define representational salience. To do so, we denote the similarity between two trials z, z' by $\text{Sim}(O(z, z')) := K(z, z')$. Note that this is well-defined for compositionally structured representations, as $K(z, z')$ is identical for all pairs of trials with the same $O(z, z')$. We then define:

Definition B.1. For a generalized compositionally structured representation with kernel K and a conjunction $J \subseteq \{1, \dots, C\}$, we recursively define

$$\overline{S}(\emptyset) := K(\emptyset), \quad \overline{S}(J) := \text{Sim}(J) - \textstyle\sum_{J' \subsetneq J} \overline{S}(J'), \quad S(J) = \frac{\overline{S}(J)}{\sum_{\emptyset \neq J \subseteq \{1,\dots,C\}} \overline{S}(J)}. \tag{14}$$

When emphasizing the total number of components, we denote $S(J; C) = S(J)$. Further, for compositionally structured representation, the similarity only depends on the total number of overlaps $|J|$. We therefore write $S(k; C) = S(J)$, where $k := |J|$.

To illustrate how salience would be computed in practice, we consider an example representation $X \in \mathbb{R}^{n \times d}$. This representation gives rise to a kernel $K = XX^T \in \mathbb{R}^{n \times n}$. We can also understand the trial-by-trial similarity in terms of the components it is overlapping in — denoting the set of all subsets of $\{1, \dots, C\}$ by $\mathcal{J}$, we denote this by $O \in \mathcal{J}^{n \times n}$. In particular, $O_{ii} = \{1, \dots, C\}$. For example, if data points 1 and 2 overlap in the third component, $O_{12} = \{3\}$. $\text{Sim}(J)$ is then defined as the entries K_{ij} where $O_{ij} = J$. Note that in a compositionally structured representation, whenever $O_{ij} = J$, K_{ij} takes on the same value, but more generally, we can define $\text{Sim}(J)$ by taking the average similarity. We can then define the unnormalized salience $\overline{S}$ by recursively computing it from $\text{Sim}(J)$ using Eq. (14). We then normalize the salience to get our final estimate.

B.2 Why is representational salience a useful metric?

Intuitively, the representational salience captures the unique contribution of a population representing a particular conjunction J (as in our informal definition in the main text). As we noted in the main text, distinguishing these unique contributions from the similarities directly would be more difficult. For example, changes in the similarity between inputs having two components in common could arise from changes in how saliently single components or conjunctions of two components are represented.

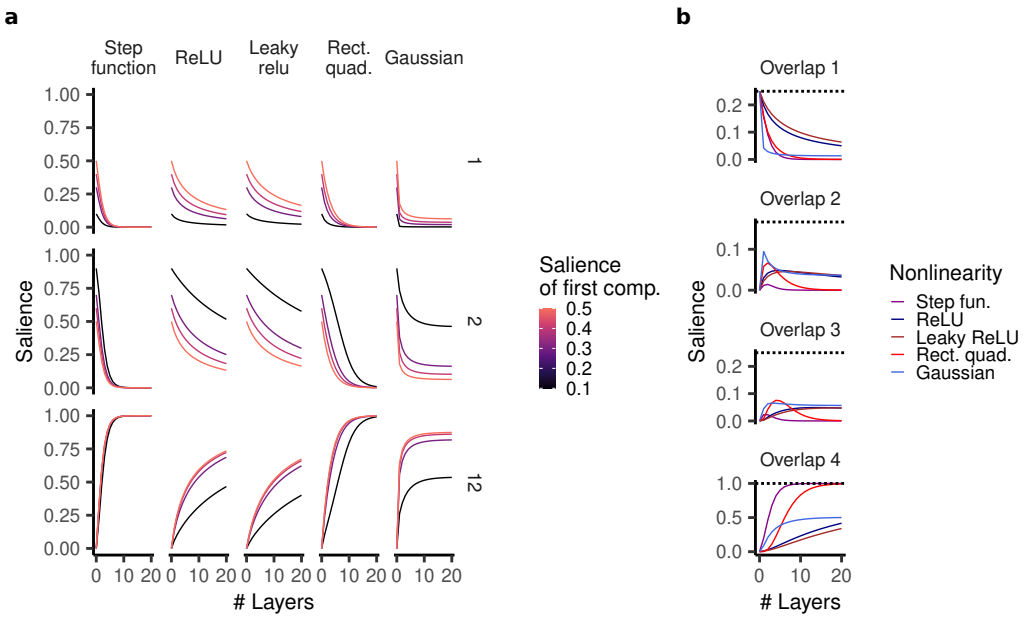

Figure 7: Extended analysis of overlap salience in random neural networks. **a**, Salience of the first component (1), the second component (2), and their conjunction (12), where we vary the two components' saliences in the input. **b**, Salience for inputs with four components.

To distinguish between these cases, we'd have to additionally look at the similarity between inputs having a single component in common. Our definition of representational salience avoids this issue; we therefore believe that this perspective could more broadly be useful for analyzing representations of compositional data.

B.3 Why does representational salience leave out magnitude and baseline activity?

Notably, many learning models are largely invariant to a constant rescaling of their representation. For limit behavior (e.g. in the limit of infinite training in gradient descent), this scale has no impact. We see this, for instance, in Proposition 5.2, where the scaling turns out to be entirely irrelevant. For regularized models, on the other hand, the scale of the representation is confounded with the strength of the regularization and so we suggest that it is again best seen as separate from the representational geometry.

The baseline activity, on the other hand, may determine how easily an intercept is learned. However, many linear readout models (e.g. support vector machines) do not regularize their intercept, again rendering this parameter entirely irrelevant. Notably, in the case of gradient descent, the magnitude of the baseline activity does influence how easily an intercept is learned. However, the impact of this is often negligible; for example, in Proposition 5.2, we again find that the baseline activity is entirely irrelevant.

B.4 Extended representational analysis

We computed the saliences by iteratively computing the representational similarities using the kernels derived in prior work (Cho & Saul, 2009; Tsuchida et al., 2018; 2019; Han et al., 2022). In Fig. 7b, we plot the different saliences for an input with four components. Now the salience of overlap 2 and 3 both first increase and then decrease and, just like for inputs with three components (Fig. 2), a Gaussian and rectified quadratic nonlinearity yields a particularly high salience for these intermediate conjunctions. Notably, they appear to be trading off the salience of these conjunctions differently: the rectified quadratic nonlinearity more strongly emphasizes overlaps of three whereas the Gaussian

nonlinearity more strongly emphasizes overlaps of two. This highlights that for larger numbers of components, the dependence of generalization behavior on specific architectural choices will likely be even stronger.

Finally, we consider an example of a generalized compositionally structured representation. Here we assume that the different input components have different magnitudes. In particular, we consider a disentangled representation whose first component has a salience between $s \in [0.1, 0.5]$ and whose second component accordingly has a salience $1 - s$ (Fig. 7a). As the random weight neural network becomes deeper, the more salient component (in this case component 2) increasingly dominates the representation. While the salience of the full conjunction still eventually converges to one for most nonlinearities, it takes longer to do so for a less balanced input representation. Further, for a Gaussian nonlinearity, an imbalanced representation actually decreases the limit salience the representation appears to be converging to for the full conjunction. This highlights that for disentangled representations that do not represent all components with equal magnitude, compositional generalization behavior may vary strongly with different neural network architectures.

B.5 Proof of Proposition 5.1

Proposition 5.1. *For a random neural network with a (leaky) ReLU nonlinearity, as $L \to \infty$, $S(k; C) \to 0$ for $k < C$ and $S(C; C) \to 1$.*

Proof. Note that the proof is a minor extension of Lemma S1.3 in Lippl et al. (2024). We present it here in a self-contained manner. We consider a nonlinearity

$$\sigma(u) := A \min(u, 0) + \max(u, 0), \quad A \in [0, 1). \tag{15}$$

By prior work (Cho & Saul, 2009; Tsuchida et al., 2018; 2019; Han et al., 2022),

$$\mathbb{E}_w \left[\phi(w^T h^{(l)}(z)) \phi(w^T h^{(l)}(z')) \right] = \sigma^2 \|h^{(l)}(z)\|_2 \|h^{(l)}(z')\|_2 k \left(\hat{h}^{(l)}(z)^T \hat{h}^{(l)}(z') \right), \tag{16}$$

where

$$k(u) = \frac{(1-A)^2}{2\pi} \left(\sqrt{1-u^2} + (\pi - \cos^{-1}(u))u \right) + Au, \tag{17}$$

σ^2 is the variance of the sampled weights, and $\hat{h} = h/\|\hat{h}\|_2$.

This means that for any two inputs that have a certain similarity u, their similarity in the L-th layer is given by $k^{(L)}(u)$, where $k^{(L)}$ denotes the L-times application of k. Let distinct trials in the input have a similarity of κ_d and let identical trials have a similarity of κ_i. Any set of trials with overlapping components will have a similarity κ, $\kappa_d < \kappa < \kappa_i$. We denote their corresponding similarity in the L-th layer by $\kappa_i^{(L)}, \kappa_d^{(L)}, \kappa^{(L)}$. Our goal is now to show that

$$\lim_{L \to \infty} s^{(L)} = 0, \quad s^{(L)} := \frac{\kappa^{(L)} - \kappa_d^{(L)}}{\kappa_i^{(L)} - \kappa_d^{(L)}} = 0. \tag{18}$$

This implies directly that the salience of all partial conjunctions converges to zero, which in turn implies that the salience of the full conjunction converges to one.

(16) implies that $\kappa_i^{(L+1)} = \sigma^2 \kappa_i^{(L)} k(1) = \sigma^2 \kappa_i^{(L)} \frac{1+A^2}{2}$. Notably, all inputs have the same magnitude and therefore have the same magnitude through all layers; this is given by $\sqrt{\kappa_i^{(L)}}$. We can therefore denote

$$\kappa^{(L+1)} = \sigma^2 \kappa_i^{(L)} k \left(\kappa^{(L)} / \kappa_i^{(L)} \right). \tag{19}$$

Thus,

$$s^{(L+1)} = \frac{k(\kappa^{(L)}/\kappa_i^{(L)}) - k(\kappa_d^{(L)}/\kappa_i^{(L)})}{k(1) - k(\kappa_d^{(L)}/\kappa_i^{(L)})} \tag{20}$$

We thus define new normalized variables $\hat{\kappa}^{(L)} := \kappa^{(L)}/\kappa_i^{(L)}, \hat{\kappa}_d^{(L)} := \kappa_d^{(L)}/\kappa_i^{(L)}$, i.e.

$$s^{(L)} = \frac{\hat{\kappa}^{(L)} - \hat{\kappa}_d^{(L)}}{1 - \hat{\kappa}_d^{(L)}}, \tag{21}$$

and therefore

$$\hat{\kappa}^{(L)} = (1 - \hat{\kappa}_d^{(L)})s^{(L)} + \hat{\kappa}_d^{(L)}. \tag{22}$$

Note that $\hat{\kappa}^{(L+1)} = k(\hat{\kappa}^{(L)})/k(1)$ and $\hat{\kappa}_d^{(L+1)} = k(\hat{\kappa}_d^{(L)})/k(1)$. We thus define

$$\tilde{k}(u) := \frac{k(u)}{k(1)} = u + \frac{\rho}{\pi}(\sqrt{1-u^2} - \cos^{-1}(u)u), \quad \rho := \frac{(1-A)^2}{(1+A)^2}. \tag{23}$$

Note that

$$\tilde{k}'(u) = 1 + \frac{\rho}{\pi}\left(-\frac{u}{\sqrt{1-u^2}} - \cos^{-1}(u) + \frac{u}{\sqrt{1-u^2}}\right) = 1 - \frac{\rho}{\pi}\cos^{-1}(u). \tag{24}$$

Note that $k(1) = 1$ and as for all $0 \leq u < 1$, $\tilde{k}'(u) < 1$, this is the only fixed point and $\hat{\kappa}^{(L)}, \hat{\kappa}_d^{(L)} \to \infty$. We can therefore define

$$s^{(L+1)} = \frac{\tilde{k}\left((1 - \hat{\kappa}_d^{(L)})s^{(L)} + \hat{\kappa}_d^{(L)}\right) - \hat{\kappa}_d^{(L)}}{1 - \kappa_d^{(L)}}. \tag{25}$$

We now determine the fixed mapping to this mapping assuming that $\hat{\kappa}_d^{(L)}$ is fixed at some value d, i.e.:

$$f(s,d) = \frac{\tilde{k}((1-d)s + d) - \tilde{k}(d)}{1 - \tilde{k}(d)}. \tag{26}$$

$s = 0$ is a fixed point. Further,

$$\frac{\partial f(s,d)}{\partial s} = \frac{(1-d)\tilde{k}'((1-d)s+d)}{1-\tilde{k}(d)} = \frac{(1-d)(1 - \frac{\rho}{\pi}\cos^{-1}((1-d)s+d)}{1 - d - \frac{\rho}{\pi}(\sqrt{1-d^2} - \cos^{-1}(d)d)}. \tag{27}$$

We now prove that this for $0 < s \leq \frac{1}{2}$, this derivative is smaller than 1. Specifically,

$$\begin{aligned}&(1-d)(1 - \tfrac{\rho}{\pi}\cos^{-1}((1-d)s+d) = \\ &1 - d - \tfrac{\rho}{\pi}(\sqrt{1-d^2} - \cos^{-1}(d)d) + \tfrac{\rho}{\pi}r(s,d),\end{aligned} \tag{28}$$

where the residual is given by

$$r(s,d) := (d-1)\cos^{-1}((1-d)s + d) + \sqrt{1-d^2} - d\cos^{-1} d. \tag{29}$$

We now need to prove that $r(s,d) < 0$. Note that $r(s,d)$ is monotonically increasing in s and therefore

$$r(s,d) \leq r(\tfrac{1}{2}, d) = (d-1)\cos^{-1}(\tfrac{1}{2} + \tfrac{1}{2}d) + \sqrt{1-d^2} - d\cos^{-1} d < 0, \tag{30}$$

where we infer the latter inequality by visual inspection of the plot of this function. □

C DSPRITES REPRESENTATIONS

To investigate whether our theory can describe compositional generalization in practically used disentangled models, we considered six different model architectures considered in Locatello et al. (2019) and trained on the DSprites dataset, an important benchmark for disentangled representation learning. This paper considers fifty random seeds for each model and further considers six different possible hyperparameters per architectures, resulting in a total of 1,800 models. In our analysis, we only analyze differences between architectures, pooling across all hyperparameter choices.

C.1 TASK SETUP

DSprites consists of small black-and-white shapes and has five different underlying components: shape (three categories: hearts, ovals, and squares), size, x-position, y-position, and rotation. We consistently consider the largest possible size and hold the rotation fixed at zero. While x- and y-position can each take on 32 possible values, we only consider four different categories for each.

Importantly, this is still a highly non-compositionally structured representation: not only do thes disentangled representation learning methods often fail to discover the underlying factors of variation; the model also likely represents x- and y-position that are closer to each other, as more similar, violating our compositionally structured assumption. These representations therefore present a particularly challenging test of our framework.

For symbolic addition and transitive equivalence, we consider x- and y-position as the two components. For each task instance, we randomly determine which position takes on which component's role. In total, we consider 50 randomly sampled task instance for each of the 1,800 models. For context dependence, we subsample two shapes and assume that these shapes provide the context cue. We then consider x- and y-position as feature 1 and feature 2. While the context dependence considered in the main text had six possible feature values for each feature, we now only consider four. As a result, we only consider two different datasets: CD-2, which leaves out all trials where feature 1 indicates category 2 and feature 2 indicates category 1; and CD-1 which only leaves out one conjunction of features. Put differently, on a given trial, the task could for example look as follows: if we see a heart shape, the model should output whether this object is in a certain x-position; if we see a square shape, the model should output whether this object is in a certain y-position. While the model has seen all x- and y-positions individually, it has not seen each combination of x- and y-positions.

Finally, we consider either a direct linear readout from the disentangled representation or an initial transformation by a one-hidden-layer ReLU neural network with random weights.

C.2 The models are well described by a conjunction-wise additive computation

We first investigated whether the model predictions on the test set were well characterized by a conjunction-wise additive computation (see Appendix D.2), averaging this model behavior across the fifty random task instances. We found that they were generally well captured, though they were certainly not perfectly conjunction-wise additive (Fig. 8a). Further, none of these models were able to systematically generalize on transitive equivalence (Fig. 8b). This indicates that conjunction-wise additivity may characterize the generalization class of these highly non-compositionally structured representations as well — at least when averaged across task instances.

C.3 The representations are partially compositionally structured

We next investigated whether these representations exhibit compositional structure according to our definition. To determine this, we computed their representational similarity matrix and computed the average similarity for each set of overlaps. This instantiates the compositionally structured representational similarity matrix that most closely described the empirical representational similarity. We then determined the average squared deviation from this matrix, σ^2_{cs}, comparing it to the overall variance of this matrix, σ^2_{var}. Overall, the *variance ratio*, $\sigma^2_{cs}/\sigma^2_{var}$, characterizes the degree to which the given representation is compositionally structured. In a fully non-compositionally structured representation, the variance ratio will be roughly one, whereas in a fully compositionally structured representation, it will be zero. For the disentangled representation learning models, we found that this variance ratio took on value between 0.5 and 0.9 (Fig. 8c). Thus, the models' representation partially captured compositional structure, but also contained a lot of non-compositional structure.

C.4 The non-compositional representational structure substantially impacts model behavior

Finally, we tested whether our representational geometry analysis in Section 5 extended to the DSprites representations. We first estimated their salience $S(1;2)$ by computing the average representational similarity for each overlap and then computing the salience using Eq. (14). Proposition 5.2 would predict that the generalization error should decrease with increasing $S(1;2)$. However, we do not observe such a trend (Fig. 8d). This indicates that our representationa geometry analysis does not apply to the DSprites representations, due to the way in which they violate the compositional structure assumption.

Finally, we determined generalization accuracy on CD-2 and CD-1. We found that most models performed quite poorly, perhaps owing to the fact that shape is barely represented in these models. Nevertheless, we found that most models performed systematically below chance for CD-2 but

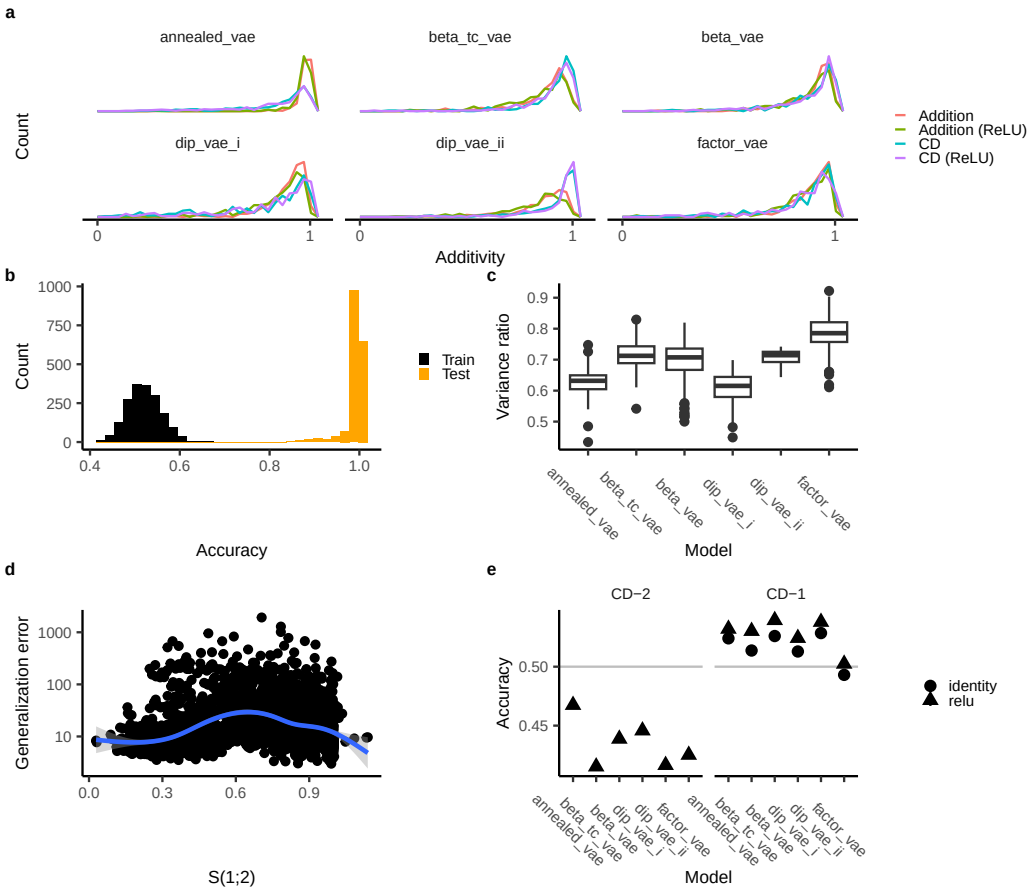

Figure 8: Conjunction-wise additivity in empirical disentangled models. **a**, Additivity across the different model architectures and four different task settings. All of the model behaviors are generally highly additive. **b**, Distribution of accuracies on transitive equivalence when averaged across different component roles. **c**, Variance ratio, capturing the ratio between the deviation between the representational similarity matrix and its compositionally structured approximation and the general variance of the representational similarity matrix. The numbers indicate that all disentangled models discover some compositional structure in their representation, but still deviate substantially from a compositionally structured representation. **d**, Generalization error on symbolic addition plotted against the estimated $S(1;2)$. **e**, Average accuracy on two versions of context dependence with a total of four features: CD-2 which now leaves out an entire orthant, and CD-1, which only leaves out a single data point. Again, models generally perform below chance on CD-2 but above chance on CD-1, though performance is generally really low.

systematically above chance for CD-1. Our analysis of context dependence in compositionally structured representations suggests that these may be because these models exploit a context-driven shortcut on CD-2 but not CD-1.

Overall, our analysis therefore paints a nuanced picture of the applicability of our theory to practical disentangled representations. Our findings suggest that these models may still be well approximated by conjunction-wise additive computations and that this generalization class may therefore shed light on fundamental limits to the generalization of linear readout models. At the same time, to understand how these models generalize on conjunction-wise additive tasks, taking into account the specific ways in which they violate the compositional structure assumption is likely important.

D Detailed methods

D.1 Models

Kernel model. We fit the kernel models by hand-specifying the kernel and fitting either a support vector regression or classification using `scikit-learn` (Pedregosa et al., 2011). Note that this is equivalent to using a feature basis with the same resulting similarities. However, hand-specifying these similarities enabled us to easily explore the full range of possible representations.

Rich and lazy ReLU networks. All networks were trained with Pytorch and Pytorch Lightning Paszke et al. (2019). We consider ReLU networks with one hidden layer and $H = 1000$ units. We initialize by $\sigma\sqrt{2/H}$, considering $\sigma \in [10^{-6}, 1]$. In particular, when reporting results on rich networks (without further specification), we assume $\sigma = 10^{-6}$. When reporting results on lazy network, we assume $\sigma = 1$.

Compositional MNIST/CIFAR-10. We create a compositional version of MNIST and CIFAR-10 by concatenating multiple images along different channels. For example, the input to the MNIST version of the symbolic addition task had two channels, each containing one digit whereas the CIFAR-10 version of the symbolic addition task had six channels, three for each digit. We further varied the distance between the concatenated images either presenting them all on top of each other or presenting them offset by a certain number of pixels. Each image category corresponded to a certain component, where its role as ranodmly sampled for each task instance. The output was still given by a single scalar: the total magnitude for symbolic addition and the two categories for context dependence and transitive equivalence.

Convolutional neural networks. We considered networks with four convolutional layers (kernel size is five, two layers have 32 filters, two have 64 filters) and two densely connected layers (with 512 and 1024 units). Each layer is followed by a ReLU nonlinearity, and the convolutional stage is followed by a max pooling operation. All weights are initialized with He initialization (He et al., 2015). We trained these networks with SGD using a learning rate of 10^{-4} and momentum of 0.9.

Residual neural networks. We trained a residual neural network with eight blocks in total, two with 16, 32, 64, and 128 channels, respectively, using the Adam optimizer with a learning rate of 10^{-3} for 100 epochs.

Vision Transformers. Finally, we trained a Vision Transformer (ViT) with six attention heads, 256 dimensions for both the attention layer and the MLP, and a depth of four, using Adam with a learning rate of 10^{-4} for 200 epochs.

Data augmentation. We did not use data augmentation for MNIST. For CIFAR-10, we used a random flip and a random crop.

D.2 Additivity analysis

To analyze how well a conjunction-wise additive computation can describe network behavior, we considered as the set of possible features a concatenation of one-hot vectors coding for each possible

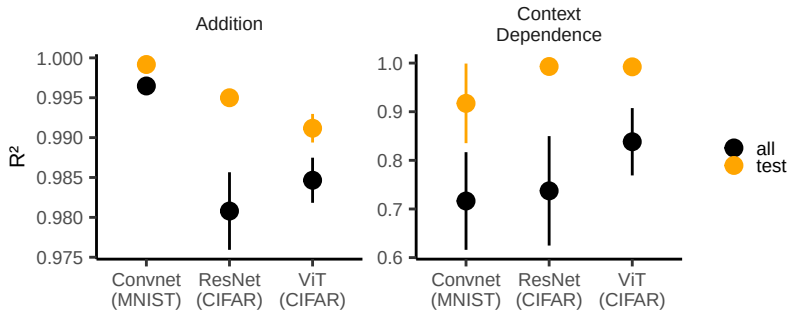

Figure 9: R^2 of the conjunction-wise additive predictor on the test set as well as on both the training and test set.

conjunction. We then removed all features that are constant at zero on the training dataset and used linear regression to try and predict network behavior on both training and test set for all remaining features. The resulting R^2 defines the "additivity" of the network behavior (i.e. $R^2 = 1$ indicates full conjunction-wise additivity). Furthermore, we can use the inferred values assigned to these different conjunctions to compare kernel models, feature-learning networks, and vision networks. Note that for the vision networks, we first average the model predictions across all different images instantiating a given compositional input.

Notably, our theory predicts that the constrained conjunction-wise additive function can predict behavior on the test set, but not necessarily on the training set. To test this prediction in vision network, we compared the additivity of the test set alone to the additivity of the entire dataset (Fig. 9). First, we found that the predictivity on the test set was indeed generally very high. Second, we found that it was substantially higher than on the training and test set taken together.

E Compositional Task Space

E.1 Symbolic Addition

E.1.1 Proof of Proposition 5.2

Proposition 5.2. *Consider inputs $z_1, z_2 \in \{[v]\}_{v\in\mathcal{V}}, \mathcal{V} \subset \mathbb{R}$, with associated values v. We assume that the training set contains all pairs such that at least one component is $z_c \in \{[w]\}_{w\in\mathcal{W}}, \mathcal{W} \subset \mathcal{V}$ and that the average value in both $\mathcal{V}$ and $\mathcal{W}$ is zero. Then, model behavior on the test set is given by*

$$f([v_1],[v_2]) = m(v_1 + v_2), \quad m := \frac{p\cdot S(1;2)}{1+(p-2)S(1;2)}, \quad p := |\mathcal{W}| \tag{4}$$

Proof. We split up the training data into

$$\mathcal{I}^{(1)} := \{[w]|w \in \mathcal{W}\}^2, \tag{31}$$

and

$$\mathcal{I}^{(2)} := \bigcup_{w\in\mathcal{W}} \mathcal{I}^{(1,w)} \cup \mathcal{I}^{(2,w)}, \tag{32}$$

$$\mathcal{I}^{(1,w)} := \{([w],[v])|v \in \mathcal{V}\setminus\mathcal{W}\}, \quad \mathcal{I}^{(2,w)} := \{([v],[w])|v \in \mathcal{V}\setminus\mathcal{W}\}. \tag{33}$$

We denote the dual coefficient associated with each training point $([i],[j])$ by $a_{ij} \in \mathbb{R}$. Note that the problem is symmetric and therefore we know that $a_{ij} = a_{ji}$. We define a few summed coefficients:

$$b_v := \sum_{w\in\mathcal{W}} a_{vw}, \quad \bar{b}_w := \sum_{v\in\mathcal{V}\setminus\mathcal{W}} a_{vw}, \quad c_w := \sum_{w'\in\mathcal{W}} a_{ww'}, \tag{34}$$

$$b := \sum_{v\in\mathcal{V}\setminus\mathcal{W}} b_v = \sum_{w\in\mathcal{W}} b_w, \quad c := \sum_{w\in\mathcal{W}} c_w. \tag{35}$$

Note that the sum over all dual coefficients is given by $2b + c$. Let $p := |\mathcal{W}|$ and $q := |\mathcal{V}| - |\mathcal{W}|$. We denote by κ_c the similarity between inputs overlapping in c components. Then, setting $\delta_2 := \kappa_2 - \kappa_0$ and $\delta_1 := \kappa_1 - \kappa_0$, the set of dual equations is given by

$$([w], [w']) \in \mathcal{I}^{(1)} : \kappa_0(2b + c) + \delta_1(\overline{b}_w + \overline{b}_{w'} + c_w + c_{w'}) + (\delta_2 - 2\delta_1)a_{ww'} = w + w', \quad (36)$$

$$([w], [v]) \in \mathcal{I}^{(1,w)} : \kappa_0(2b + c) + \delta_1(b_v + \overline{b}_w + c_w) + (\delta_2 - 2\delta_1)a_{wv} = w + v. \quad (37)$$

(Note that the equation corresponding to $([v], [w])$ is equivalent due to the problem's symmetry.)

The prediction is given by

$$f([v_1], [v_2]) = \kappa_0(2b + c) + \delta_1(b_{v_1} + b_{v_2}). \quad (38)$$

We now sum (37) over w (setting $\overline{w} := \sum_{w \in \mathcal{W}} w$):

$$\begin{aligned} \overline{w} + pv &= p\kappa_0(2b + c) + p\delta_1 b_v + \delta_1(b + c) + (\delta_2 - 2\delta_1)b_v \\ &= ((p - 2)\delta_1 + \delta_2)b_v + (2p\kappa_0 + \delta_1)b + (p\kappa_0 + \delta_1)c \end{aligned} \quad (39)$$

Thus,

$$\delta_1 b_v = \frac{\delta_1 p}{(p - 2)\delta_1 + \delta_2} v + \delta_1(\overline{w} - (2p\kappa_0 + \delta_1)b - (p\kappa_0 + \delta_1)c). \quad (40)$$

Thus, setting

$$m := \frac{\delta_1 p}{(p - 2)\delta_1 + \delta_2}, \quad d := 2\delta_1(\overline{w} - (2p\kappa_0 + \delta_1)b - (p\kappa_0 + \delta_1)c) + \kappa_0(2b + c), \quad (41)$$

we can write

$$f([v_1], [v_2]) = m(v_1 + v_2) + d. \quad (42)$$

To simplify d, we sum (36) over all w, w':

$$\begin{aligned} 2p\overline{w} &= p^2\kappa_0(2b + c) + 2p\delta_1(b + c) + (\delta_2 - 2\delta_1)c \\ &= 2p(p\kappa_0 + \delta_1)b + (p^2\kappa_0 + 2(p - 1)\delta_1 + \delta_2)c. \end{aligned} \quad (43)$$

We further sum (39) over all v, setting $\overline{v} := \sum_{v \in \mathcal{V} \setminus \mathcal{W}} v$:

$$((p + q - 2)\delta_1 + \delta_2 + 2pq\kappa_0)b + q(p\kappa_0 + \delta_1)c = q\overline{w} + p\overline{v}. \quad (44)$$

We can now compute b and c from this system of equations and plug it into d.

Finally, $\overline{w} = \overline{v} = 0$ immediately implies that $b = c = 0$ and therefore $d = 0$. □

E.1.2 Behavior of deep ReLU networks

Next, we examined the behavior of deep ReLU networks with varying depth (Fig. 10). On the extrapolation task ($\mathcal{W} = \{0\}$), these networks also had compressed model predictions on the compositional split that were approximately linearly distorted (though for deeper networks, this relationship seemed to have a slight S-shape). Intriguingly, these networks perfectly generalized on the interpolation task ($\mathcal{W} = \{-4, 4\}$). For $\mathcal{W} = \{-2, 2\}$, they perfectly generalized on the interpolation portion, but not the extrapolation portion.

E.1.3 Behavior of vision models

We trained the Convnets on MNIST using seven different training sets $\mathcal{W}$: $\{[0]\}$, $\{[-4], [4]\}$, $\{[-2], [2]\}$, $\{[-4], [0], [4]\}$, $\{[-1], [0], [1]\}$, $\{[-4], [-3], [3], [4]\}$, $\{[-2], [-1], [1], [2]\}$. This allowed us to vary both the size of the training set and whether it involved only interpolation or also extrapolation. We trained these networks for 100 epochs on 20,000 samples. We trained the ResNets and ViTs on CIFAR-10 using the first three training sets listed above. We trained the ResNets for 100 epochs and the ViTs for 200 epochs, training all networks on 40,000 samples. Our findings on these experiments are summarized in Section 6. Further, Fig. 11 depicts, for each training set, the average prediction for each combination of components. We can see that the ConvNet predictions are highly linearly correlated with the ground truth. The ResNet and ViT predictions are also linearly correlated, but exhibit a slightly noisier relationship.

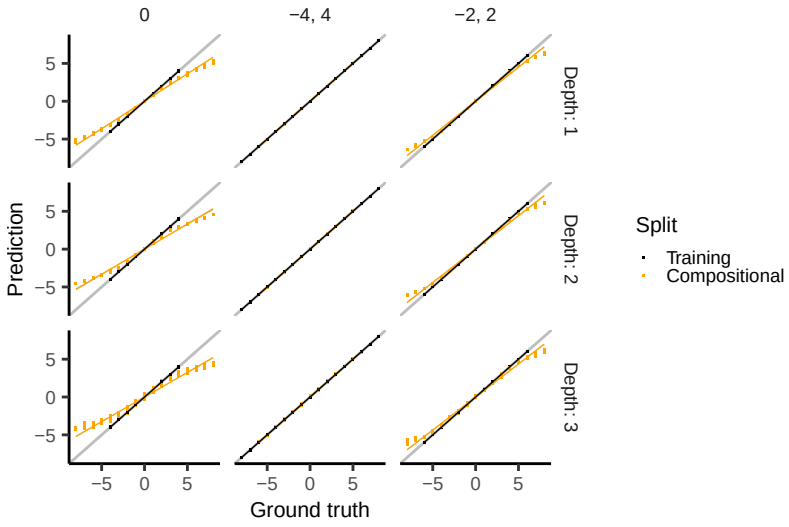

Figure 10: Model prediction plotted against ground truth for different training sets and depths.

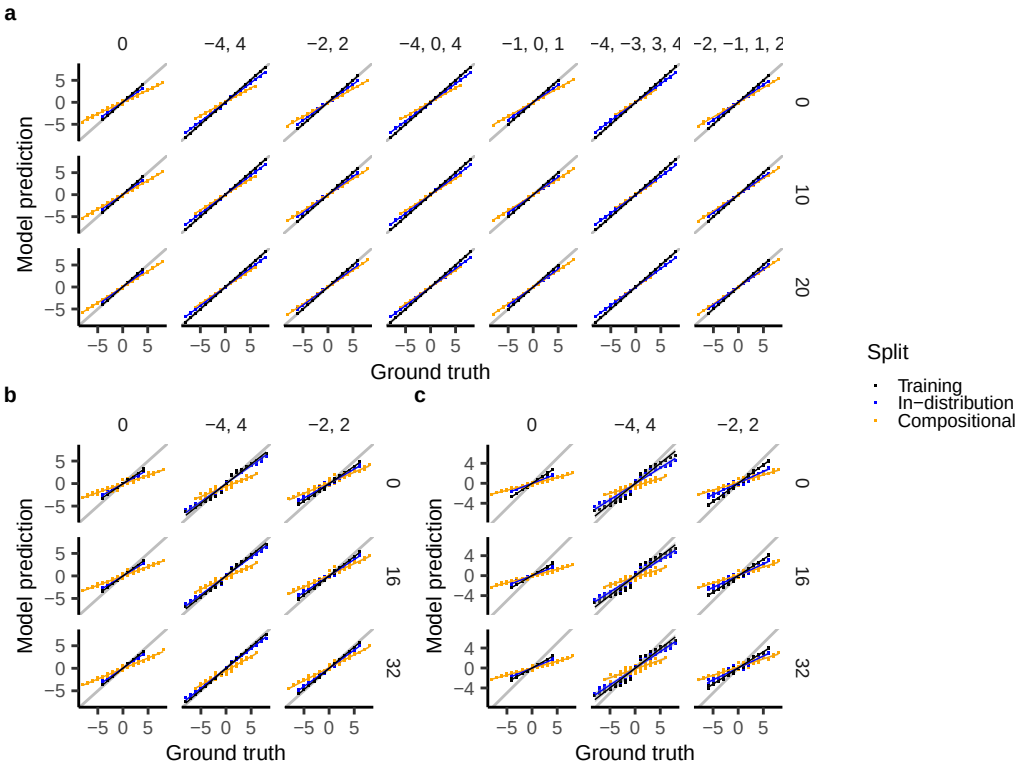

Figure 11: Average prediction for each combination of components plotted against the ground truth for **a**, ConvNets trained on MNIST, **b**, ResNets trained on CIFAR-10, and **c**, ViTs trained on CIFAR-10.

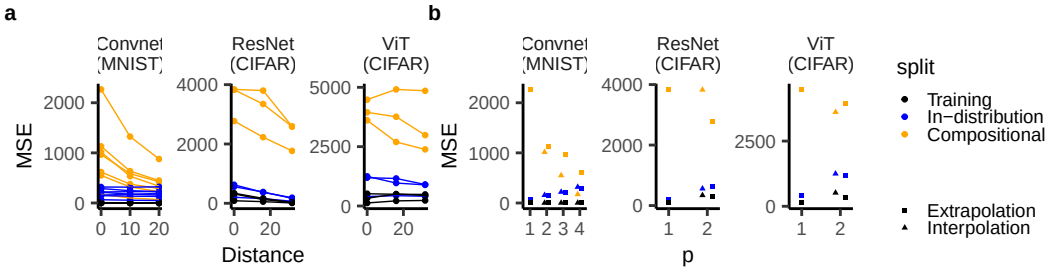

Figure 12: **a**, Mean squared error as a function of distance. **b**, Mean squared error as a function of training set size p (different dots corresponding to interpolation versus extrapolation.

Figure 13: **a**, $S(1;2)$ in the convolutional neural network's neural tangent kernel, before and after being trained on symbolic addition with $\mathcal{W} = \{0\}$. **b**, The estimated slope for each model instance plotted against $S(1;2)$. While increased $S(1;2)$ generally yields a larger slope (as predicted by our theory), the relationship between $S(1;2)$ and the slope does not follow the exact quantitative relationship predicted by our theory.

Additionally, we also plot the mean squared error of the networks on the different training sets (Fig. 12). This provides a more conventional (but harder to interpret) measure of performance compared to the slope depicted in Fig. 4. We again see that the mean squared error on the compositional generalization set is decreasing with increasing distance (Fig. 12a). Further, it is decreasing with increasing training set size (Fig. 12), though we note that the differences in the scales of the generalization set for different sets $\mathcal{W}$ renders it harder to directly to compare these values. In contrast, the slopes described in the main text are more easily comparable.

Overall, our analysis indicates that our theory provides several valuable qualitative insights into the behavior of deep neural network models. Importantly, however, we are not able to provide quantitative bounds at the moment. To demonstrate this, we computed the salience $S(1;2)$ of the neural tangent kernel of the convolutional neural networks before and after being trained on the MNIST version of symbolic addition with $\mathcal{W} = \{0\}$ (Fig. 13a). Importantly, $S(1;2)$ changed substantially during training, indicating that these networks are not trained in the kernel regime. As such, our theory does not apply exactly. To see whether it provided appropriate quantitative bounds, we then estimated the slope of each model instance on the compositional generalization dataset and plotted it against $S(1;2)$ of the neural tangent kernel at the beginning of training. We found that higher $S(1;2)$ generally resulted in a larger slope, as predicted by our theory. However, the relationship between $S(1;2)$ and slope did not adhere exactly to the quantitative predictions made by our theory. This indicates that while our theory can shed light on certain qualitative behaviors, it still leaves certain generalization behaviors in deep neural networks unclear.

Finally, our analysis in Section 5.1 suggests that deeper models should yield more conjunctive representations, and therefore exhibit worse generalization on symbolic addition. To test this prediction, we varied the number of fully connected layers in the convolutional neural network and trained these networks on the symbolic addition task with the training set $\mathcal{W} = \{0\}$. We found that deeper networks indeed generalized worse (Fig. 14).

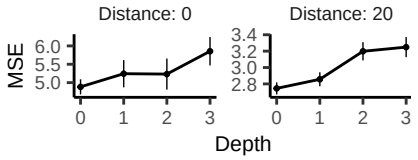

Figure 14: Mean squared error on the compositional generalization set for different depths of the fully connected network at the end of the convolutional neural network architecture.

E.2 Context dependence

E.2.1 General task definition

We consider inputs with three components, $(z_{\text{co}}, z_{f1}, z_{f2})$. We assume that $z_{\text{co}} \in C_1 \cup C_2$, where C_1 is the set of possible contexts under which z_{f1} is relevant and C_2 is the set of possible contexts under which z_{f2} is relevant. We further assume that there are decision functions $d_1(z_{f1}), d_2(z_{f2}) \in \mathbb{R}$. (For example, in the example in the main text, these function map three features to the first category (i.e. $y = -1$) and three features to the second category (i.e. $y = 1$).) The target is then given by

$$y(z_{\text{co}}, z_{f1}, z_{f2}) = \begin{cases} d_1(z_{f1}) & \text{if } z_{\text{co}} \in C_1, \\ d_2(z_{f2}) & \text{if } z_{\text{co}} \in C_2. \end{cases} \tag{45}$$

Note that in the main text, we consider $C_1 = \{1\}$, $C_2 = \{2\}$, and six possible values for z_{f1}, z_{f2}, where the decision function maps three onto 1 and three onto -1.

E.2.2 Novel stimulus compositions are conjunction-wise additive

If the test set consists in novel combinations of stimuli, this is a conjunction-wise additive computation. Namely, suppose that for all test inputs $(z_{\text{co}}, z_{f1}, z_{f2})$, the two features have never been observed in conjunction, but both (z_{co}, z_{f1}) and (z_{co}, z_{f2}) have been. (This includes the case considered in the main text.) In this case, we can define functions f_{12} and f_{13} to implement the appropriate mapping:

$$f_{12}(z_{\text{co}}, z_{f1}) := \begin{cases} d_1(z_{f1}) & \text{if } z_{\text{co}} \in C_1, \\ 0 & \text{if } z_{\text{co}} \in C_2, \end{cases} \quad f_{13}(z_{\text{co}}, z_{f2}) := \begin{cases} 0 & \text{if } z_{\text{co}} \in C_1, \\ d_2(z_{f2}) & \text{if } z_{\text{co}} \in C_2, \end{cases} \tag{46}$$

$$f(z_{\text{co}}, z_{f1}, z_{f2}) = f_{12}(z_{\text{co}}, z_{f1}) + f_{13}(z_{\text{co}}, z_{f2}). \tag{47}$$

E.2.3 Novel rule compositions are not conjunction-wise additive

We could also imagine an alternative generalization rule in a task where there are multiple components indicating the same context: $C_1 = \{\text{co}_1, \text{co}_2\}$ and $C_2 = \{\text{co}_3, \text{co}_4\}$. We then leave out certain features with certain contexts. For example, suppose we had never seen two values for z_{f1} and z_{f2} in conjunction with $z_{\text{co}} \in \{\text{co}_2, \text{co}_4\}$. In principle, if the model understood that $z_{\text{co}} = \text{co}_1, \text{co}_2$ (and $z_{\text{co}} = \text{co}_3, \text{co}4$ resp.) signify the same context (i.e. learned to abstract the context from the context cue), it could generalize successfully as it had observed these features in conjunction with $z_{\text{co}} = \text{co}_1, \text{co}_3$. However, the conjunction-wise additive mapping depends on having observed each context in conjunction with each feature and this task is therefore non-additive.

E.2.4 Coefficient groups

In Fig. 3d, we grouped the inferred coefficients into categories. We here explain these categories:

- Right conj.: This is the correct conjunction the model should use to solve the task, i.e. between $z_{\text{co}} = \text{co}_1$ and z_{f1} and between $z_{\text{co}} = \text{co}_2$ and z_{f2}.
- Wrong conj.: This is the incorrect conjunction between context and feature, i.e. between $z_{\text{co}} = \text{co}_1$ and z_{f2} and between $z_{\text{co}} = \text{co}_2$ and z_{f1}.
- Sensory feat.: This is any conjunction involving sensory features, i.e. $z_{f1}, z_{f2}, (z_{f1}, z_{f2})$.
- Context only: This is the component z_{co} by itself.

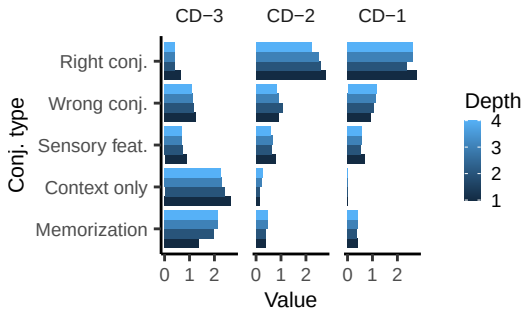

Figure 15: Inferred values for different conjunctive groups on context dependence.

- Memorization: This is the full conjunction of all three components $(z_{\text{co}}, z_{f1}, z_{f2})$.

We then compute the average absolute magnitude within each of these groups in order to determine their overall relevance to model behavior.

E.2.5 Feature-learning ReLU networks

We find that feature-learning ReLU networks generalize consistently on *CD-1* and CD-2 but not *CD-3*. They are also perfectly conjunction-wise additive and fail due to a context shortcut (Fig. 15).

E.2.6 Vision networks

We additionally analyzed ConvNets for different distances between the digits. We found that convolutional neural networks trained on MNIST successfully generalized on *CD-1* and *CD-2*, but not CD-3 (Fig. 16). For smaller distances between digits, the models tended to generalize worse on *CD-1* and *CD-2* and gradually reverted to chance accuracy (i.e. 0.5) for *CD-3*. Further, the networks were generally highly additive ($R^2 > 0.95$), but became worse for lower distance (Fig. 16b). Finally, across all distances, they had a high magnitude associated with the context cue, though this magnitude decreased for small distances — consistent with the accuracy of the network increasing from below chance to chance level (Fig. 16c).

E.3 Invariance and partial exposure

We consider invariance and partial exposure as simple case studies for the memorization leak and shortcut bias. In both cases, the input consists of two components and the mapping only depends on the first (Fig. 17a). In the invariance case, we don't see the second component vary at all, in the partial exposure case, we see one instance of the second component. Note that the partial exposure task has previously been studied in the context of network generalization (Dasgupta et al., 2022; Chan et al., 2022).

To understand the impact of different representational saliences, we consider the generalization margin $m := y\hat{y}$ on the test set, where y is the ground-truth label and $\hat{y}$ is the model's estimate. Because we consider support vector machines, the margin on the training set is one; a smaller margin on the test set indicates worse performance. We determined a mathematical formula for the margins as a function of $S(1;2)$. Below we first describe its implications and then how we derived this.

Invariance suffers from a memorization leak. On invariance, we find that the model's test margin is expressed as $m = \frac{S(1;2)}{1-S(1;2)}$. This means that for a fully compositional representation ($S(1;2) = 0.5$), its training and test margins are both one. However, as $S(1;2)$ decreases, the model increasingly memorizes the training set, resulting in a decreased margin (Fig. 17b).

Shortcut distortion. On the partial exposure task, if the model used item 1 to solve the task, it would get two out of three training examples correct and could memorize the last data point. This is a statistical shortcut and we find that norm minimization (just as for context dependence) ends up

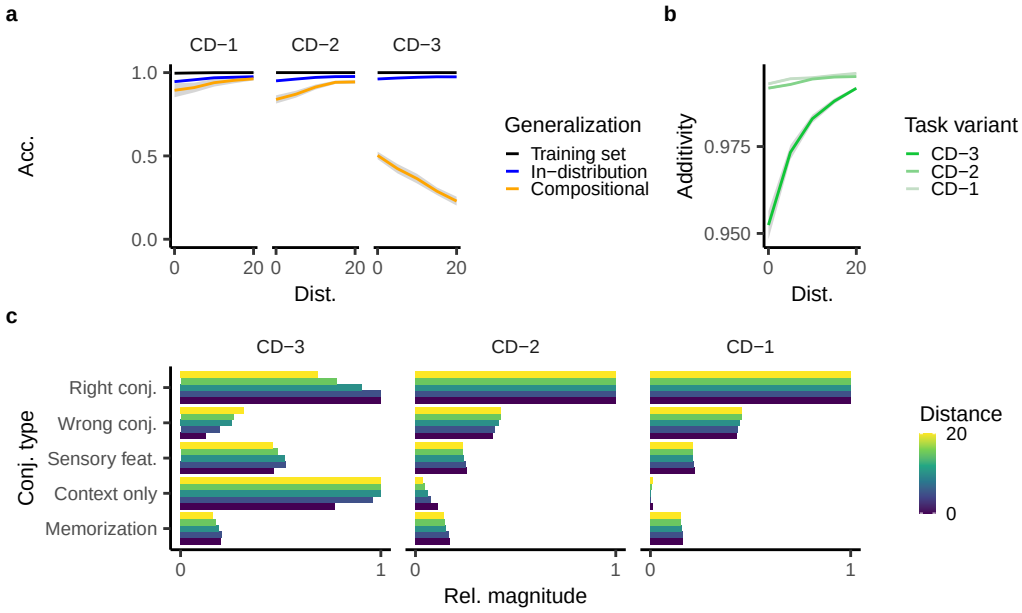

Figure 16: Performance of convolutional neural networks trained on an MNIST version of context dependence. For different distances and training sets, we plot **a**, the accuracy on different splits, **b**, the additivity of the networks, and **c**, the magnitude of the different inferred coefficients.

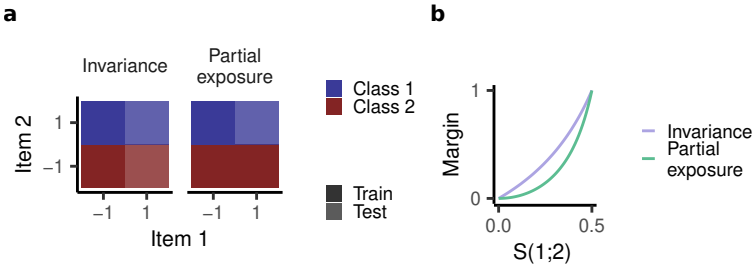

Figure 17: **a**, Schema for invariance and partial exposure. **b**, Generalization margin for the two tasks.

partially relying on this strategy as this decreases the ℓ_2-norm of the readout weights. As a result, the test margin for the partial exposure task decreases even more strongly as a function of $S(1;2)$: $m = \frac{2S(1;2)^2}{1-2S(1;2)^2}$ (where we assume that the similarity between identical trials is one and the similarity between distinct trials is zero) (Fig. 17b).

Derivations. We analytically compute the kernel models' test set prediction on the invariance task. The training set is given by $\{(-1,-1),(-1,1)\}$ and its kernel is therefore

$$K = \begin{pmatrix} \kappa_2 & \kappa_1 \\ \kappa_1 & \kappa_2 \end{pmatrix}, \tag{48}$$

where κ_2 is the similarity between identical trials and κ_1 is the similarity between overlapping trials. Hence, the dual coefficients are given by

$$a = K^{-1}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\kappa_2^2 - \kappa_1^2}\begin{pmatrix} \kappa_2 & -\kappa_1 \\ -\kappa_1 & \kappa_2 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \frac{1}{\kappa_2^2 - \kappa_1^2}\begin{pmatrix} \kappa_2 + \kappa_1 \\ -(\kappa_2 + \kappa_1) \end{pmatrix}. \tag{49}$$

The test set is given by $\{(1,-1),(1,1)\}$ and its kernel with respect to the training set is therefore

$$\tilde{K} = \begin{pmatrix} \kappa_1 & \kappa_0 \\ \kappa_0 & \kappa_1 \end{pmatrix}, \tag{50}$$

where κ_0 is the similarity between distinct trials. Hence the test set predictions are given by

$$\hat{y} = \tilde{K}a = \frac{1}{\kappa_2^2 - \kappa_1^2} \begin{pmatrix} (\kappa_2 + \kappa_1)(\kappa_1 - \kappa_0) \\ -(\kappa_2 + \kappa_1)(\kappa_1 - \kappa_0). \end{pmatrix} \tag{51}$$

As the ground truth labels are $y = \{1, -1\}$, the margin $m = y\hat{y}$ is identical for both test set points:

$$m = \frac{(\kappa_2 + \kappa_1)(\kappa_1 - \kappa_0)}{\kappa_2^2 - \kappa_1^2} = \frac{\kappa_1 - \kappa_0}{\kappa_2 - \kappa_1} = \frac{(\kappa_2 - \kappa_0)S(1;2)}{(\kappa_2 - \kappa_0) - (\kappa_1 - \kappa_0)} = \frac{S(1;2)}{1 - S(1;2)}. \tag{52}$$

For partial exposure, the training set is given by $\{(-1,-1),(-1,1),(1,-1)\}$ and its kernel is therefore

$$K = \begin{pmatrix} \kappa_2 & \kappa_1 & \kappa_1 \\ \kappa_1 & \kappa_2 & \kappa_0 \\ \kappa_1 & \kappa_0 & \kappa_2 \end{pmatrix}. \tag{53}$$

The test set is given by $\{(1,1)\}$ and the test set kernel is therefore

$$\tilde{K} = \begin{pmatrix} \kappa_0 & \kappa_1 & \kappa_1 \end{pmatrix} \tag{54}$$

The margin is therefore given by

$$m = y\hat{y} = -\hat{y} = -\tilde{K}K^{-1} \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}. \tag{55}$$

We solve this equation for the special case where $\kappa_0 = 0$ and $\kappa_1 = 1$ using Mathematica and find that

$$m = \frac{2S(1;2)^2}{1 - 2S(1;2)^2}. \tag{56}$$

E.4 Other mathematical operations

We could consider mathematical operations other than addition as well, considering unobserved assigned values $v_1(z_1)$ and $v_2(z_2)$ together with some composition function $C(v_1(z_1), v_2(z_2))$. This task will only be additive if the composition function is additive (e.g. if it is subtraction). If it is, e.g. multiplication, division, or exponentiation, the task will be non-additive.

E.5 Logical operations

In this task, inputs with two components are presented. Each component z_c has an unobserved truth value $T(z_c)$ associated with it and the target is some logical operation over these two truth values, for example *AND*: $T(z_1) \wedge T(z_2)$. After inferring the truth value of each component, the model could generalize towards novel item combinations. As long as the logical operation is additive (e.g. *AND*, *OR*, *NEITHER*, ...), this is an additive task. If the logical operation is non-additive (e.g. *XOR*), this would be a non-additive task. Indeed, the *XOR* case is structurally equivalent to the transitive equivalence task.

E.6 Transitive ordering

Transitive ordering is a popular task in cognitive science (often called transitive inference, McGonigle & Chalmers (1977)). Here the subject is presented with two items z_1, z_2 drawn from an unobserved hierarchy $>$. It should then categorize whether $z_1 > z_2$ or $z_2 > z_1$. Crucially, this task can be solved by assigning a rank $r(z_c)$ to each item and computing the response as $f(z) = r(z_1) - r(z_2)$ Lippl et al. (2024). It is therefore additive, in contrast to transitive equivalence. This is also the case if we assume that there are multiple such hierarchies (e.g. $a_1 > \ldots, a_5$ and $b_1 > \ldots, b_5$). In this case, the model would generalize to comparisons between these different hierarchies as well.

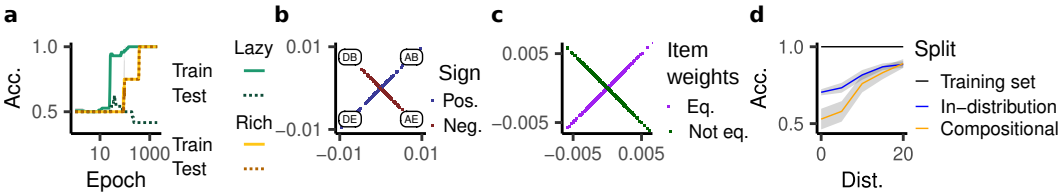

Figure 18: Feature learning enables generalization on transitive equivalence. **a**, Learning curves in the lazy and rich regime. **b**, The weights of the network in the subspace corresponding to one underlying *XOR*-task. **c**, Weights for the same unit are plotted against each other and colored by whether they correspond to equivalent items (purple) or non-equivalent items (green). **d**, Accuracy of a ConvNet trained on an MNIST version of transitive equivalence.

F A FEATURE-LEARNING MECHANISM FOR NON-ADDITIVE COMPOSITIONAL GENERALIZATION

We proved above that compositionally structured kernel models do not generalize on non-additive tasks, including transitive equivalence. To see if feature learning can overcome this limitation, we trained ReLU networks through backpropagation on transitive equivalence, using a disentangled and uncorrelated input. By varying the initial weight magnitude, we either trained these networks in the kernel/lazy regime or the feature-learning/rich regime. Notably, when trained on symbolic addition and context dependence, the rich networks were well-described by a conjunction-wise additive model (Figs. 10 and 15). On transitive equivalence, however, while the kernel-regime models failed to generalize (as predicted by our theory), the rich networks generalized correctly (Fig. 18a).

To explain why this is the case, we leveraged the insight that rich neural networks are biased to learning weights with a low overall ℓ_2-norm (Lyu & Li (2020); Chizat & Bach (2020); cf. Vardi & Shamir (2021)). In particular, a one-hidden-layer ReLU network tends to learn a sparse set of features Savarese et al. (2019); Chizat & Bach (2020). Transitive equivalence consists of multiple overlapping equality relations (e.g. $A = B \neq D = E$). Notably, ReLU networks such an *XOR*-type problem by specializing one unit to each conjunction (Brutzkus & Globerson (2019); Saxe et al. (2022); Xu et al. (2023); Fig. 18b). Further, their sparse inductive bias incentivizes ReLU networks to use identical units for overlapping conjunctions (e.g. $(A, B), (A, C)$, and (B, C)). This causes the unit to generalize to unseen item combinations (e.g. (A, C)), enabling the network to generalize. Importantly, our theoretical argument is corroborated by empirical simulations: each network unit has identical weights for equivalent items (Fig. 18c).

Thus, rich networks' capacity for abstraction gives rise to an additional compositional motif, allowing them to generalize on transitive equivalence. In particular, our findings highlight that transitive equivalence and transitive ordering are solved by fundamentally different network motifs.

To see whether large-scale neural networks can also benefit from this feature-learning mechanism, we trained ConvNets on an MNIST version of transitive equivalence. The networks were trained for 150 epochs on 20,000 samples. We found that if the digits were presented with a distance of zero, the network did not generalize compositionally at all. However, with increasing distance, the network started to improve its compositional generalization (Fig. 18d), demonstrating that a convolutional network can benefit from this rich compositional motif.

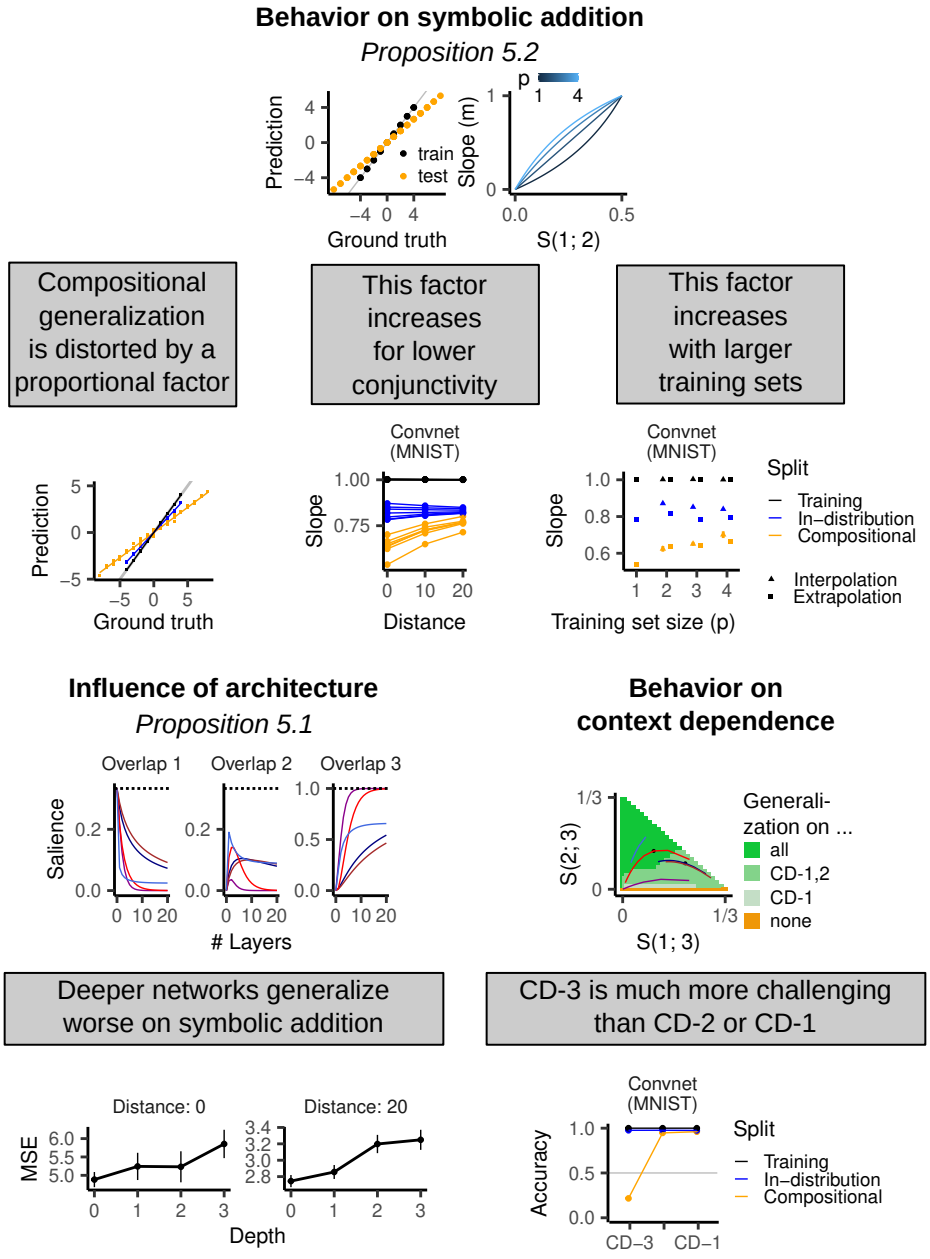

Figure 19: This figure composes the relevant theoretical insights and empirical experiments to illustrate how they speak to each other.